MASTER EN SCIENCES INFORMATIQUES

Étude de la réalisation d'une infrastructure à tiers de confiance

Léonard, Benjamin

Award date:
1997

Awarding institution:
Universite de Namur

Etude de la réalisation d'une infrastructure à tiers de confiance

B. Léonard

Mémoire réalisé par Benjamin Léonard

en vue de l'obtention du diplôme de
« Licence et Maitrîse en Informatique »,

Promoteur : Monsieur Jean Ramaekers

Année académique 1996-97

Facultés Universitaires Notre-Dame de la Paix

INSTITUT D'INFORMATIQUE

Rue de bruxelles 61 - 5000 NAMUR

Tel.081/72.41.11 - Telex 59222 Facnam-b - Telefax 081/23.03.91

Etude de la réalisation d'une infrastructure à tiers de confiance

LEONARD Benjamin

Résumé

Dans ce travail, nous présentons les notions de cryptographie nécessaires à la réalisation d'une infrastructure à tiers de confiance. Une fois ces notions introduites, les différents rôles des tiers de confiance seront mis en évidence. Nous étudierons, dans un second chapitre, plusieurs protocoles de communication. Ensuite, nous analyserons, par le biais des spécifications fonctionnelles, une infrastructure à tiers de confiance basée sur le protocole à clefs de session « amélioré » vu au chapitre deux. Nous verrons alors quelques problèmes liés à la légalité des tiers de confiance. Nous finirons ce travail par la présentation d'infrastructures à tiers de confiance existant en Belgique.

Abstract

In this work, we present some knowledges in cryptography needed to make of Trusted Third Party's structure. Once this done, we inlightened the different parts of a TTP.We will study , in the second chapter, some communication protocol. Then, we will analyse a TTP structure, based on the communication protocole with session keys of the previews chapter, via a fonctionnals requirements. We will speak about some issue of the legality related with TTP. We will finish with the presentation of some existing TTP in Belgium.

Mémoire de licence et maîtrise en science informatique

septembre 97

Promoteur: . J.Ramaekers

Je commencerai par remercier Monsieur le professeur Jean Ramaekers qui m'a proposé ce travail et a bien voulu le promouvoir. Il m'a notamment aidé, le long de cette année, à rassembler la documentation et définir les grandes lignes.

Je remercierai également Pascal Goossens pour ses nombreux conseils et le temps qu'il a passé à m'aider pour la rédaction, ainsi que Etienne Davio

Je remercierai encore mes nombreux correcteurs : Laurent, Bénédicte, Janina et les autres.

Je finirai par remercier les étudiants pour la bonne ambiance qu'ils ont maintenue pendant mes deux années à l'institut.

TABLE DES MATIÈRES

Table des matières

INTRODUCTION

Introduction

Actuellement, il est bien connu qu'Internet est un réseau mondial, mais que ce réseau est peu sécurisé. Il est également de plus en plus reconnu que la cryptographie peut contribuer à la sécurité de communications critiques comme celles issues du commerce électronique, de la santé ou de la vie privée. La cryptographie ne représente qu'une facette de cette sécurisation. Entre autres, beaucoup de protocoles de communication se sont basés sur la présence de tiers de confiance pour sécuriser les communications. Ce travail est centré sur ces tiers de confiance. Nous parlerons des outils cryptographiques utiles aux tiers de confiance, de certains protocoles de communication, pour enfin spécifier une infrastructure de tiers de confiance.

Dans le cadre du commerce électronique, il est intéressant de voir les attentes des utilisateurs. Michael Froomkin [FROO96] parle de ces attentes et divise les utilisateurs en deux catégories : les vendeurs et les acheteurs. Les principaux désires du vendeur sont :

- L'authentification : le vendeur doit connaître l'identité de l'acheteur avant de conclure la vente afin de prouver le bon de commande et de garantir le paiement. Il peut également désirer réaliser une base de données reprenant sa clientèle.
- La certification : le vendeur a parfois besoin de vérifier certaines caractéristiques de l'acheteur avant de conclure la vente (par exemple, si la vente requiert un âge minimum).
- La confirmation : le vendeur doit pouvoir prouver à une autre entité (comme une compagnie de cartes de crédit) que l'acheteur a bien autorisé le paiement.
- La non répudiation de l'origine : le vendeur veut être protégé des clients qui prétendraient de façon injustifiée ne pas avoir passé de commande.

Les désirs du client sont relativement proches de ceux du vendeur :

- L'authentification : la confirmation de l'identité du vendeur est importante en cas de litige sur la qualité de la marchandise ou en cas de services

après-vente.

- L'intégrité : elle consiste à la protection contre des paiements non autorisés.

Essayons de voir comment ce travail permet d'aider à réaliser une infrastructure TTP répondant à ces desiderata. Dans un premier temps, nous avons commencé par expliquer le principe du chiffrement asymétrique qui permet la signature électronique utile pour remplir la plupart des desiderata cités supra. Ensuite, nous avons déterminé les différents rôles que peuvent tenir les tiers de confiance. Nous avons explicité les différents rôles qui sont ceux de l'enregistrement, de la certification, de la gestion des certificats et de la génération des paires de clefs asymétriques. Après cette section, le lecteur aura une bonne idée de comment les différentes autorités répondent aux attentes des utilisateurs. D'autres techniques annexes ont encore été développées.

Dans la suite de ce travail, nous proposerons plusieurs protocoles qui répondent selon les cas à certaines attentes des utilisateurs. Nous avons alors choisi un de ces protocoles et nous avons développé une infrastructure basée sur ce dernier dans le chapitre 3. Cette infrastructure permet l'authentification, la non-répudiation et la confirmation. La certification n'est pas prévue mais pourrait être facilement ajoutée.

Le quatrième chapitre ne traite plus directement des tiers de confiance mais nous y énonçons des problèmes légaux concernant la cryptographie. Ces problèmes sont en rapport direct avec les tiers de confiance. Nous parlons donc dans ce chapitre de la légalité du chiffrement face aux lois sur les écoutes téléphoniques (déjà un peu développée au chapitre 1) et de la signature électronique.

Le dernier chapitre de ce travail va reprendre deux exemples de tiers de confiance existant sur le marché belge : Isabel et BelSign dont on pourrait s'inspirer pour un environnement plus général que celui étudié dans le cadre de ce travail.

Mais commençons par introduire ces notions dont nous avons besoin pour comprendre les tiers de confiance.

LA CRYPTOGRAPHIE

1. La cryptographie

1.1 Introduction

Le début de ce chapitre a un but essentiellement didactique. Il vise à rappeler quelques principes de cryptographie. Dans cette première partie, nous fixerons des notions de vocabulaire et des notations qui reviendront régulièrement dans ce travail.

Dans un second temps, nous vous parlerons plus précisément des possibilités du chiffrement asymétrique. Nous découvrirons ce qu'est la signature électronique et la façon de préserver la confidentialité d'un document au moyen d'une paire de clefs. Nous dirons alors deux mots sur la génération des nombres aléatoires. Ensuite, nous énoncerons les principes de trois algorithmes de chiffrements asymétriques.

Dans la suite du chapitre, nous parlerons donc de ce qui est le centre de ce travail : les tiers de confiance. Nous énoncerons les différents rôles des tiers de confiance. Mais nous ne développerons pas l'ensemble de ces rôles car certains seront rencontrés dans le chapitre suivant.

Pour finir ce chapitre, nous énoncerons des techniques cryptographiques qui nous semblent relativement importantes pour la suite de ce travail.

1.2 Les principes de la cryptographie

Essayons, pour commencer, de voir l'utilité de la cryptographie en sécurité des systèmes informatiques. Nous savons qu'assurer la sécurité pour des communications et des stockages d'informations par des moyens physiques est hors de prix si nous n'utilisons pas des systèmes centralisés et isolés (dits systèmes fermés). Avec le développement d'Internet, le besoin de sécurité se fait de plus en plus ressentir. Car il n'est pas très compliqué pour un tiers malveillant d'accéder à une information ou de se faire passer pour

quelqu'un d'autre dans des réseaux où il n'y a pas de contrôle, comme le réseau des réseaux qu'est Internet. L'importance de la cryptographie paraît alors évidente puisque c'est un moyen de contrôler si une information a été modifiée ou de rendre confidentielle une information. Mais ceci est réalisé en attachant la sécurité au message lui-même et non au matériel.

Nous donnerons la définition suivante de la cryptographie: La cryptographie est la science et l'étude de l'écriture secrète [DENN82].

On peut dire également que c'est un moyen de représenter une information de façon telle qu'elle ne soit compréhensible qu'à l'aide d'un jeu de clefs, ou encore une manière de rendre une information inutilisable à quiconque ne possède pas la méthode de déchiffrement.

a) Question de vocabulaire

Faisons dès à présent de petites mises au point au niveau du vocabulaire.

Dans une communication, il y a toujours un **émetteur** (A) et un **récepteur** (B). Ici, l'émetteur désire que personne ne puisse affecter le message de quelque façon que ce soit. Pour réaliser ce désir, l'émetteur va modifier le **texte en clair** (P:plaintext) pour arriver à un **texte chiffré** (C:cyphertext). Le processus de modification du texte en clair pour cacher sa substance est appelé le **chiffrement**. Le processus inverse est appellé le **déchiffrement**. Le **décryptage** peut être défini comme l'art de décrypter les textes chiffrés (modifier le texte chiffré afin d'obtenir, sans les clefs, le texte original). La branche des mathématiques qui regroupe ces sciences s'appelle la **cryptologie**. Pour clarifier ces idées, un schéma résumant ces notions se trouve à la figure 1.

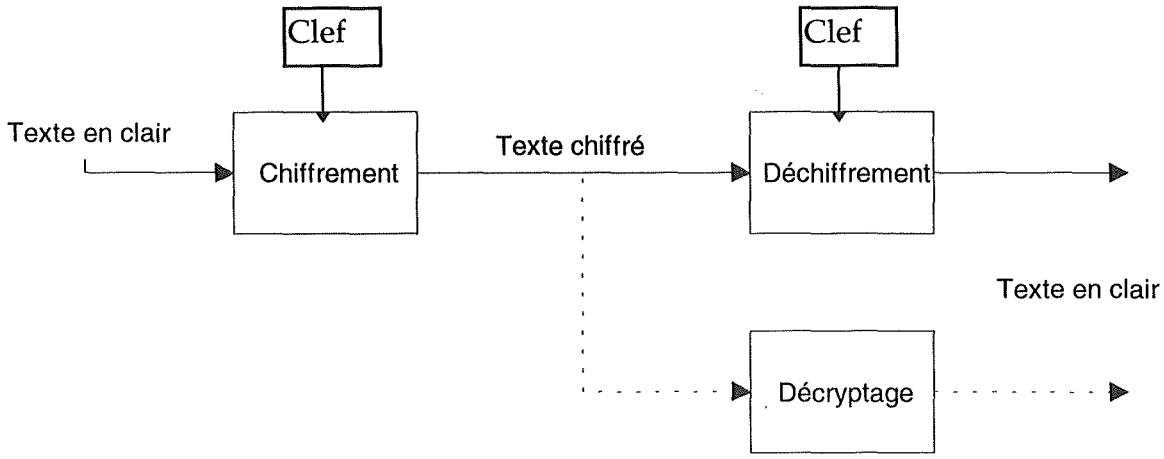

Figure 1. Définition: Chiffrement, Déchiffrement et Décryptage

b) L'algorithmique

[SCHN93],[DENN82]

Pour chiffrer et déchiffrer un texte, un algorithme est utilisé. Il est souvent basé sur des propriétés mathématiques (ex.: théorème d'Euclide). On appelle **algorithme restreint** l'algorithme dont le secret réside dans l'algorithme lui-même. Ce type d'algorithme n'a plus que des intérêts historiques. Il est plus aisé de garder un secret dans une clef que dans un algorithme, pour des raisons évidentes de programmation.

A l'heure actuelle, les algorithmes utilisent des clefs (k) dont la valeur appartient à **un espace des clefs** et affecte les fonctions de chiffrement. De là, viennent d'autres notions telles que les algorithmes de chiffrement symétrique qui possèdent une même clef pour le chiffrement et pour le déchiffrement. L'algorithme le plus utilisé pour ce type de chiffrement s'appelle le DES [RAEM94],[SCHN93]. Nous noterons le chiffrement et le déchiffrement :

$$E_k(P)=C$$
$$D_k(C)=P$$

P étant le texte en clair et C le texte chiffré.

Mais l'utilisation des algorithmes symétriques limite les possibilités de la cryptographie. Car deux entités vont utiliser la même clef pour chiffrer et déchiffrer les messages entre elles. Cela permet la confidentialité. Mais, par

contre, il est impossible de discerner l'émetteur d'un message parmi les deux entités. Une des deux entités pourrait prétendre que l'autre entité lui a envoyé un message, personne ne pourrait savoir si le message vient de l'autre entité ou vient d'elle-même. Ce problème a été résolu avec l'avènement d'un autre type d'algorithme où le chiffrement se réalise avec une clef et le déchiffrement grâce une autre clef. Grâce à ce principe, il est possible de déterminer l'origine d'un message, les protagonistes d'une conversation utilisant des clefs de chiffrements différentes.

On utilisera pour ces algorithmes asymétriques une notation différente. Ceux-ci permettent en plus du chiffrement, la signature d'un document.

$E_{pk}(P)=C \Rightarrow$ Chiffrement de P avec une clef publique
$D_{sk}(C)=P \Rightarrow$ Déchiffrement de C avec une clef secrète
[1] $S_{sk}(P)=C' \Rightarrow$ Signature de P par un chiffrement avec une clef secrète
$V_{pk}(C')=P \Rightarrow$ Vérification d'une signature par le déchiffrement de C avec une clef publique

Donnons encore deux dernières définitions avant de clore cette partie et de parler de la génération de nombres aléatoires.

Un **protocole de communication** est défini, dans [SCHN93], comme étant une séquence d'étapes, impliquant au moins deux parties, en vue de réaliser une communication . Un **cryptomodule** est un élément hardware utilisé en cryptographie. Par exemple, une carte à puce est un cryptomodule.

---

[1] Dans le cas d'une signature , il y a deux possibilités: soit on chiffre tout le message; soit on applique une fonction de hachage sur le message , on chiffre le résultat et on joint le tout au message. Le récepteur peut alors réaliser le même hachage sur le message et comparer avec ce qu'il a reçu et déchiffrer grâce à la clef publique.

c) La génération de séquence de nombres aléatoires et pseudo-aléatoires

La cryptographie est très sensible aux propriétés des générateurs de nombres aléatoires. Si un générateur de nombres aléatoires est prévisible, un tiers malveillant qui arrive à connaître la suite d'une séquence de nombres aléatoires pourra alors tromper le système. Ce problème peut devenir important si le protocole de communication utilise les nombres aléatoires pour éviter le rejeu (réception d'un même message deux fois) ou si le protocole les utilise dans un processus d'identification. Un exemple de cette deuxième possibilité est expliqué dans les commentaires du protocole X.509 au chapître suivant. Il est évident que la génération de nombres aléatoires est également essentielle lors de la génération des clefs de chiffrement.

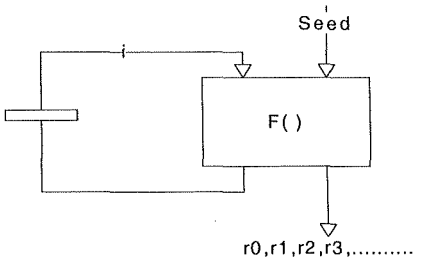

Figure 2. Générateur de nombres aléatoires: La fonction F calcule R0 en fonction d'une valeur introduite (seed) et ensuite calcule les valeurs suivantes en fonction de chaque résultat.

Si nous utilisons uniquement des fonctions analytiques implémentées dans un ordinateur, il paraît peu probable que nous obtenions des séquences entièrement aléatoires. Nous citerons une phrase du mathématicien Von Neuman pour justifier cette assertion: ' Anyone who considers arithmetical of producing random digits is, of course, in a state of sin'. Nous pouvons cependant générer des séquences de nombres que l'on appelle pseudo-aléatoires. Ces dernières sont générées par des fonctions arithmétiques et semblent aléatoires (elles passent tous les tests statistiques). La plupart des générateurs aléatoires peuvent être représentés sous la forme du schéma repris dans la figure ci-dessus.

Ces séquences de nombres sont de toute façon périodiques car si nous considérons le générateur comme sur la figure ci-dessus , et que la fonction

est surjective, alors la même séquence se reproduira après un certain temps à cause du paradigme des anniversaires (où la probabilité θ de retrouver deux fois le même nombre dans une population de n en prenant un échantillon de p est égale à $\theta = 1 - \frac{n!}{(n-p)! \times n^p}$ et croît très vite). Nous retrouverons une séquence comme dans la figure ci-dessous.

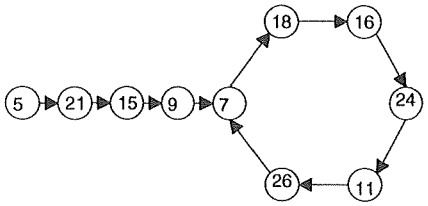

Figure 3. Séquence de nombres aléatoires où la fonction est $Y_i = (\exp(3*Y_{i-1}))$ mod 29 et seed = 5

Les applications de chiffrement demandent à avoir des séquences pseudo-aléatoires qui sont « plus » aléatoires que pour d'autres applications. « Cryptographiquement » aléatoire ne veut pas dire seulement statistiquement aléatoire, mais cela inclut également qu'un suivant est imprévisible par rapport à son précédent dans une séquence aléatoire.

Des séquences de chiffres purement aléatoires sont de niveau philosophique ou quantique. Mais nous pouvons dire qu'un générateur est « real-world random» si en lançant deux fois ce générateur avec les mêmes données en entrée (avec les limitations humaines), nous obtenons des séquences différentes.

Comme exemple, on pourra citer PGP qui utilise des intervalles de frappes au clavier pour calculer des clefs.

1.3 Les systèmes de chiffrement à clefs publiques

a) Introduction

Le chiffrement à clefs publiques est un concept inventé en 1976 par Diffie et Hellman . Un de leurs articles [DIFF 76] a motivé le développement de

systèmes similaires. Notamment Rivest, Shamir, Adleman [RIVE 78] [RIVE 88] ont développé un algorithme plus connu sous le nom de RSA. Mais beaucoup d'autres systèmes ont vu le jour. Nous citerons encore El Gamal, qui a été adopté dans une norme américaine.

Dans le chiffrement asymétrique, une procédure de chiffrement unique par utilisateur, que nous noterons $E_{user}( )$, est distribuée à tous les utilisateurs. Chaque utilisateur préserve sa procédure de déchiffrement $D_{user}()$. Ces procédures sont caractérisées par les propriétés suivantes :

1) D_{user} permet le déchiffrement d'un message P chiffré à l'aide de la procédure E_{user} :D_{user} (E_{user} (P))=P
2) Les deux procédures doivent être faciles à calculer car elles seront utilisées souvent et donc il faut limiter le temps de calcul. A l'heure actuelle, un autre facteur est à considérer : c'est la taille des algorithmes (D_{user} ( ) et E_{user} ( )) car si ceux-ci sont de taille réduite, ils peuvent être intégrés dans des cartes à puce[WALL90].
3) La révélation de E_{user} ( ) à tous les utilisateurs ne doit pas permettre à ceux-ci de trouver facilement D_{user} ( ).
4) Les procédures sont aussi applicables dans le sens différent, c'est-à-dire E_{user} (D_{user} (P))=P. On déchiffre d'abord et puis on chiffre (Cette propriété n'est pas nécessairement respectée).

Généralement, la méthode de chiffrement consiste en une procédure principale et une paire de clefs de chiffrement. Un message chiffré reste sécurisé tant que la clef de déchiffrement ne reste connue que du destinataire. Le secret réside donc dans cette clef de déchiffrement ou clef secrète. De même pour révéler E_{user} ( ), il suffit de distribuer la clef de chiffrement appelée la clef publique.

La première propriété est la propriété de chiffrement et de déchiffrement. La deuxième propriété permet de conserver le secret dans des cryptomodules (éléments hardware de chiffrement). La troisième propriété se traduit par le fait qu'il est « impossible » de déterminer la clef privée à partir de la clef publique. Enfin, la dernière propriété permet à l'utilisateur de signer son message, car lui seul est capable de chiffrer une information

avec son algorithme secret et tous les autres peuvent alors prouver l'origine de cette information en la déchiffrant avec son algorithme public. Pour signer son message, l'utilisateur peut également utiliser une fonction de hachage dont il chiffre le résultat. Une fonction de hachage est une fonction qui s'applique sur l'entièreté du texte, qui paraît presque injective et que l'on ne peut pas inverser. De plus, le résultat de l'application de cette fonction est de taille limitée par rapport à l'argument.

La confidentialité

La cryptographie est un bon moyen de rendre un message confidentiel. Le destinataire d'un message est le seul à connaître la procédure de déchiffrement pour obtenir le message en clair . Toute personne écoutant (espionnant) la ligne de communication ne peut rien tirer de l'information volée (C := « cyphertext »).

L'intérêt pour ces techniques est donc relativement important, notamment pour les banques, les états, les hôpitaux et tout se qui touche à la vie privée des personnes. En 1975, « The National Bureau of Standard » a adopté une de ces techniques comme standard, le DES (Data Encryption Standard). Le problème de ces techniques classiques de chiffrement est la distribution sécurisée des clefs (souvent résolue par un envoi postal). Ceci n'est plus vrai avec les systèmes à clefs publiques, puisque la clef public peut être mise à la disposition de tout-un-chacun dans un fichier public, n'importe qui peut envoyer un message à un tiers A, que lui seul sera capable de déchiffrer. On peut aussi imaginer que A envoie sa clef publique à B, en vue de commencer une communication. Un troisième cas de figure peut se présenter dans lequel A envoie à B un message chiffré avec la clef publique de A, qui contient une clef commune de type D.E.S. .

La signature

La signature électronique a pour but de remplacer les signatures existant sur papier. La personne, qui reçoit un message, a la preuve de l'identité de

l'émetteur (identification) et de l'authenticité du message (authentification). Mais il est aussi capable de prouver l'origine de ce message à une tierce personne si la relation entre la clef privée et l'émetteur est prouvée (non répudiation de l'origine). Ces signatures électroniques doivent dépendre à la fois de l'émetteur et du message envoyé. Dans le cas contraire, le récepteur peut changer le message ou encore simuler un autre message. Grâce à la propriété 4 , il est possible de réaliser un tel système avec le chiffrement à clef publique.

$A \Rightarrow B : (E_B(S_A(P)),A)$ où $S_A(P)$ est la signature de A qui n'est autre que le déchiffrement $D_A(P)$.

A envoie à B un message que seul B peut déchiffrer (confidentialité) et que seul A a pu écrire grâce à son algorithme de déchiffrement secret.

$B : V_A(D_B (E_B(S_A(P))))=P$

B ne peut modifier P en P' car dans ce cas la signature $V_A(P')$serait différente, d'où la notion de non-répudiation que nous introduirons plus loin.

La signature est également très utile pour prévenir des fraudes quant à la distribution des clefs publiques de chiffrement. Si ces clefs sont signées et donc certifiées par un tiers alors tous les utilisateurs sont sûrs des clefs publiques et de leur propriétaire.

Soulignons que si le message n'est pas confidentiel il est possible de signer uniquement le message.

$A \Rightarrow B : S_A(P),A$

$B : V_A(S_A(P))=P$

b) Exemples de chiffrement à clefs publiques

L'algorithme Knapsack

[SCHN93],[HELL79]

Celui-ci n'a plus qu'une utilité historique car il a été « cassé » par Shamir et Zippel [SHAM83]. Nous citerons juste un exemple d'utilisation de cet algorithme. Il fait partie des premiers algorithmes asymétriques.

Nous commençons par choisir une série de nombre appelée « superincreasing knapsack sequence » : (2, 3, 6, 13, 27, 52). Ensuite, nous multiplions chacun de ces nombres par n, qui n'a pas de facteur en commun avec les éléments de la séquence, et nous appliquons la fonction modulo m, où m est un nombre plus grand que tous les éléments de la séquence. Par exemple 31 et 56 ⇒ 2*31 mod 56 = 6. Ce qui nous donne (6, 37, 18, 11, 53, 44) qui est une série « Knapsack ».

Le fonctionnement du chiffrement est décrit dans le tableau ci-dessous. La première ligne représente la séquence pour le chiffrement. La deuxième représente le message à envoyer. Et la somme de la dernière ligne est le message chiffré.

Knapsack	6	37	18	11	53	44
Message	0	1	1	0	0	0
Somme=55	0	37	16	0	0	0

Tableau 1 : exemple de chiffrement knapsack

Pour déchiffrer le message, if faut calculer la clef de déchiffrement n*. Cette clef de déchiffrement est issue de la formule : n*×n=1 mod m qui donne pour notre exemple n*=47. L'algorithme de déchiffrement est très simple, il suffit de calculer 55×47 mod 56 = 48 que l'on traduit en binaire par 011000. Nous avons retrouvé le message initial. A l'origine, cet algorithme ne possédait pas de méthode pour la signature avec les mêmes clefs.

L'algorithme Rivest Shamir Adleman.

Cette technique (RSA) a été publiée en 1978 (RIVE 78) et a connu depuis bien des utilisations notamment dans les programmes PGP, PGP/MIME, ou ISA software.

Pour chiffrer un message P avec la méthode RSA, on utilise une clef de chiffrement issue de deux nombres (e, n) .

La première étape consiste à représenter P, le message à envoyer, comme un entier ou une séquence d'entier m de 0 à n-1. On peut utiliser des standards comme le code ASCII. Le but de cette première étape n'est pas de

chiffrer, mais de représenter sous une forme numérique nécessaire au chiffrement, le message initial.

Le chiffrement: $C \equiv E(M) \equiv M^e \bmod n$ (M étant un nombre représentant une partie du message)

Le déchiffrement : $M \equiv D(C) \equiv C^d \bmod n$

On peut remarquer que le chiffrement n'augmente pas la taille du message. Le problème qui reste est de choisir ces nombres naturels (d, e, n) de telle manière que la méthode fonctionne. Il faut d'abord calculer n comme un produit de deux nombres premiers $n = p \times q$. Ces deux nombres premiers sont relativement grands et choisis grâce à un générateur de nombres pseudo-aléatoires (C.F.2.2.3) . Comme n fait partie de la clef publique, il sera connu de tout le monde mais pas p et q qui vont servir pour déterminer e et d .

Ensuite, nous choisissons d de telle façon que le plus grand commun diviseur entre d et $\{(p-1)\times(q-1)\}$ soit égal à 1 et e est calculé à partir de d, p et q grâce à la formule : $e = \dfrac{1 \bmod((p-1)\times(q-1))}{d}$ où 1mod(x) signifie m×x+1 avec m entier quelconque.

Les algorithmes de El Gamal

[SCHN93],[ELGA85]

El Gamal utilise deux algorithmes différents : l'un pour signer et l'autre pour chiffrer (Dans l'article initial, seule la signature existait).

Sa clef publique est composée de trois nombres : y ( ), g, p (nombre premier assez grand) ou $y = g^x \bmod p$

Sa clef publique privée est :x .On remarque qu'il n'est pas possible de trouver la clef privée à partir de la clef publique. Dans le choix de ces nombres, il est important de vérifier que g<p et x<p.

La Signature

Pour signer un message M, nous choisissons un nombre aléatoire k qui reste secret et qui ne fait pas partie des diviseurs de p-1 (ou $\gcd(k, p-1) = 1$).

Grâce à ces deux nombres, il est possible de générer la signature composée de a et b où

$$a = g^k \bmod p$$
$$M = (xa + kb) \bmod p$$

Donnons un exemple de signature pour M=5, k=9, x=8, g=2, p=11, $y=2^8 \bmod 11=3$. Nous pouvons calculer $a = 2^9 \bmod 9 = 6$ et b qui est dans l'équation $5 = (6 \times 8 + 9 \times b) \bmod 10$. Nous pouvons résoudre cette équation grâce au théorème d'Euclide étendu car nous avons $(5+n \times 10) \bmod 10 = (6 \times 8 + 9 \times b) \bmod 10$ que l'on peut exprimer $n \times 10 - 9 \times b = 43$ avec n et b des entiers minimaux. On trouve alors b=3.

Pour vérifier la signature, nous disposons de M, g, p, y et bien sûr de la signature a, b . Il suffit alors de vérifier $(y^a a^b) \bmod p = g^M \bmod p$. Remarquons que nous n'utilisons ni l'élément aléatoire k ni la clef privée x. Nous nous retrouvons bien dans les caractéristiques du chiffrement asymétrique.

Vérifions pour notre exemple $3^6 \times 6^3 \bmod 11 = 10 = 2^5 \bmod 11$.

Le chiffrement

Le chiffrement n'est pas dans l'article d'origine. Il a été rajouté par la suite. Pour chiffrer un message, il faut de nouveau choisir un nombre k qui nous permet de calculer le texte chiffré composé de deux nombres a et b.

$$a = g^k \bmod p$$
$$b = y^k M \bmod p$$

Pour le déchiffrement, il suffit de calculer M avec la fonction ci-dessous :

$$M = \left\{ \frac{b}{a^x} \right\} \bmod p$$

1.4 Les tiers de confiance

a) Introduction

Pour l'instant, nous avons trouvé le moyen d'assurer la confidentialité et

l'authentification des informations. Il nous reste toujours un problème pour assurer l'identification de l'émetteur d'un message. Il nous faut un moyen de prouver que seul l'émetteur a été capable de confectionner le message chiffré avec une clef secrète. La question première est de se demander s'il est possible au récepteur (B) de prouver à un quidam quelconque(Q) que l'émetteur (A) lui a envoyé le message. Si A et B ont échangé des clefs publiques et que A a signé son message, B pourra être sûr de l'identité de A, mais rien ne prouve à Q que la clef utilisée pour vérifier la signature est bien celle correspondant à la clef secrète de A. Même la signature de A ou de B sur la clef et l'identité ($Ssk_A(A, p_{kA})$) ne suffit pas pour prouver que la signature est celle de A. Il faut alors qu'une autre entité, en qui Q a confiance, ait préalablement certifié que la clef est bien celle de A. On appelle une telle entité un tiers de confiance. La notion de **tiers de confiance** s'est développée pour rassembler plusieurs rôles différents. Ces différents rôles sont parfois inclus dans une même autorité ou au contraire assumés par des autorités différentes.

b) Les rôles des tiers de confiance

Nous avons donc séparé les différents rôles des tiers de confiance. Pour mieux situer ces rôles, nous les avons séparés dans différentes sections.

La première section rencontrée joue le rôle **d'autorité d'enregistrement**. La fonction de cette section se résume à vérifier les références (nom, adresse,.....) d'un souscripteur et à s'assurer que la clef publique fournie par le souscripteur correspond à une clef privée qui est en sa possession. Une fois ces vérifications effectuées, le résultat sera communiqué à une **autorité de certification**.

Une seconde section fondamentale est appelée autorité de certification. L'autorité de certification va vérifier les informations fournies par l'autorité d'enregistrement et ajouter des informations utiles pour décrire le profil de la certification. Elle crée alors un certificat contenant ces informations et la clef publique du souscripteur. Elle signe ce certificat avec sa clef privée. Elle va alors faire valider le certificat auprès de l'utilisateur. Ensuite, le certificat est

envoyé à l'autorité de distribution des certificats ou autorité de gestion des certificats.

L'autorité de distribution des certificats va stocker les certificats et les mettre à la disposition des utilisateurs. C'est également cette autorité qui va gérer la révocation, la suspension ou l'expiration des certificats. Elle fournit également des fonctionnalités qui permettent de vérifier des certificats périmés ou de donner des informations en cas de litige (archivage).

Les autorités suivantes sont facultatives. Il s'agit de **l'autorité de génération des clefs** qui réalise une paire de clefs asymétriques qu'elle place sur un support physique. Mais le développement actuel permet à l'utilisateur de disposer de tels systèmes sur son ordinateur personnel. Une autorité supplémentaire est **l'autorité de time stamping** qui permet d'enregistrer des preuves temporelles de la création d'un document.

Remarquons que dans la littérature, l'ensemble de ces autorités est parfois repris sous le nom unique d'autorité de certification.

c) Les autorités d'enregistrement

[PARI96]

La récolte des informations destinées à figurer dans les certificats est réalisée par l'autorité d'enregistrement (comme pour BelSign ou EDIRA). L'autorité d'enregistrement est alors chargée d'assurer un lien formel entre une personne et une paire de clefs asymétriques et de communiquer ces informations aux autorités de certification.

En règle générale, l'obtention de ces informations, et plus particulièrement celles relatives à l'identité, s'effectue en personne. En pratique, la personne qui désire enregistrer sa clef publique auprès d'une entité de certification doit se rendre au bureau de l'autorité d'enregistrement correspondante afin d'y présenter les pièces justifiant son identité, par exemple un passeport ou une carte d'identité nationale. Parfois, on rencontre d'autres moyens de vérifier l'identité comme des vérifications online avec des bases de données existantes ou la demande de garantie à une autre entité reconnue (par exemple une banque). L'autorité demande également au

souscripteur de prouver qu'il possède la clef privée correspondant à sa clef publique. Lorsque l'autorité d'enregistrement est satisfaite de l'identification, elle communique l'identité et la clef publique aux autorités de certification.

A ce stade, l'on comprend que l'efficacité du processus d'identification de même que la responsabilité de l'autorité d'enregistrement qui procède à cet exercice, représentent des éléments déterminants. En fait, la sécurité d'une infrastructure de certification repose essentiellement sur l'intégrité des informations recueillies au départ.

d) Les autorités de certification

[FROO96]

Une autorité de certification (CA: Certification authority) est une entité, soit publique soit privée, qui remplit un rôle de tiers de confiances en produisant des certificats digitaux qui attestent de certains faits concernant une personne (le souscripteur du certificat). Les certificats reprennent les faits et sont signés électroniquement par les autorités de certification. L'autorité de certification va alors faire valider le certificat auprès de l'utilisateur. Elle envoie un exemplaire du certificat au souscripteur, qui après avoir vérifier les informations contenues dans celui-ci, renvoie un message qui spécifie s'il accepte le certificat.

Les certificats sont des ensembles de données digitales que produisent des centres de certification. Les faits enregistrés dans ces certificats ont été préalablement vérifiés par une autorité d'enregistrement. Le centre de certification y ajoutera d'autres informations afin d'améliorer la gestion des certificats. Par exemple, il ajoutera les dates de validation ou les algorithmes de chiffrement utilisés par les entités concernées par ce certificat.

En pratique, le centre de certification va proposer un ensemble de certificats, classés selon le niveau « d'enquêtes » utilisées par l'autorité d'enregistrement pour confirmer l'identité du souscripteur du certificat. Prenons l'exemple de VériSign (ou son équivalent belge BelSign) qui propose quatre (trois pour BelSign) types de certificats à ses clients.

- Certificats de classe 1: Ceux-ci sont destinés au web browsing et à

l'utilisation de E-mails sécurisés. Ils certifient simplement qu'un E-mail ou un nom est identifiant dans la base de données du tiers de confiance. Ils sont obtenus par une demande du souscripteur via E-mail.

- Certificats de classe 2: Ceux-ci incluent la preuve du nom par une tierce partie.
- Certificats de classe 3: Le souscripteur doit se présenter en personne ou présenter des garants de son identité préalablement enregistrés par le CA.
- Certificats de classe 4: Ces certificats seront réalisés après une enquête complète sur le souscripteur.

Les certificats utilisés par les autorités de certification sont des certificats dits d'identification, qui lient un nom à une clef publique. Il est clair que pour les communications digitales, le nom ne doit pas être nécessairement unique ou même un nom réel. De tels certificats ne sont pas obligatoirement stockés sur des ordinateurs reliés à Internet mais peuvent être stockés sur des cartes à puce.

D'habitude un certificat comporte :

un attribut décrivant le niveau de certification
l'identité du centre de certification
un Time stamp (la date d'émission)
le nom, l'identité du souscripteur
la clef publique du souscripteur
la signature du CA

Le risque que la vérité exprimée par un certificat soit périmée peut être contrôlé, mais pas éliminé, par le fait qu'un certificat est daté quand il est produit. De plus, un certificat peut avoir une période de validité ou même le souscripteur du certificat peut périodiquement faire valider à nouveau le certificat (obtenir un nouveau certificat pour les mêmes faits auprès du CA). L'autorité de distribution des certificats doit également tenir à jour une liste de certificats révoqués.

Il existe un format de certificat bien répandu: le format X509 [SCHN93] [AZIZ96]. Il est utilisé par plusieurs autorités de certification existantes:

BELSIGN, VERISIGN.

Une autorité de certification assigne à chacun de ces utilisateurs un nom unique et crée alors un certificat x509 dont le format est:

```
Certificate-x509 ::= SIGNED SEQUENCE (
    signature                   AlgorithmIdentifier,
    issuer                      Name,
    validity                    Validity ::=SEQUENCE (
        notBefore                   UTCTime,
        notAfter                    UTCTime)
    subject                     Name,
    subjectPublicKeyInfo
    SubjectPublicKeyInfo::=SEQUENCE (
        algorithm                   AlgorithmIdentifier,
        subjectPublicKey            BIT STRING))
```

Remarquons que ce certificat permet l'utilisation de méthodes de chiffrement différentes. Le problème de l'identification du nom pourrait être résolu par une autorité particulière appellée autorité de nomination mais, nous laisserons cette tâche à l'autorité d'enregistrement.

e) Les autorités de distribution des certificats

[PARI96]

L'autorité de distribution des certificats obtient directement les certificats qu'elle distribue de l'autorité de certification. Elle pourra distribuer ces certificats de manière dynamique (à la demande) ou de manière statique en les plaçant dans des répertoires.

Puisque ces certificats contiennent des informations inaltérables, ceux-ci peuvent également être publiés dans un répertoire accessible à tous les utilisateurs. Dans ce scénario, l'émetteur n'a pas besoin d'envoyer son certificat avec son message. Le récepteur n'a qu'à consulter le répertoire dépositaire des certificats. Ce répertoire atteste que le certificat est toujours en vigueur et n'a pas été révoqué ou suspendu.

Les certificats sont généralement valides pour une période. Ils peuvent être suspendus ou révoqués pour différentes raisons dont le décès du

signataire, la modification de la raison sociale d'une entreprise, la terminaison des activités d'une entreprise, la divulgation non autorisée de la clef privée, ou plus simplement à la demande du signataire. Une révocation peut se faire par l'autorité de distribution des clefs sans l'autorisation du signataire. L'autorité de distribution des clefs pourra aussi tenir un répertoire des certificats révoqués. Pour la distribution dynamique, nous demanderons au lecteur de se référer aux différents protocoles de communication du chapître suivant.

L'autorité de distribution des clefs doit archiver différentes informations relatives aux certificats. Ces archives sont différentes de celles contenues dans le répertoire où sont publiés les certificats. Elles peuvent être conservées par un tiers et ne pas être disponibles au grand publique. Ces archives peuvent être utilisées lorsque survient un litige portant sur l'identité d'un signataire. A cette occasion, l'autorité de distribution des clefs doit pouvoir démontrer que l'autorité d'enregistrement a procédé de façon satisfaisante à l'identification de la personne désignée par le certificat et à la vérification des autres informations qui y figurent. Nous venons de parler, entre autre, de la non répudiation de l'origine qui est une propriété qui permet d'éviter qu'un émetteur (signataire) prétende qu'il n'a pas envoyé un message précis. L'autorité de gestion de certificat doit, de plus, pouvoir démontrer qu'elle a procédé de façon diligente à la publication des certificats révoqués et suspendus. Elle a enfin l'obligation de prouver que ces suspensions ou révocations ont été réalisées avec l'autorisation des signataires ou le cas échéant, qu'elle a agit de son propre chef sur la base de motifs raisonnables.

En général, l'autorité de certification et l'autorité de distribution des clefs ne font qu'un.

f) La création des clefs

Les clefs asymétriques destinées à la réalisation et à la vérification d'une signature numérique peuvent être fournies (ce qui est le cas dans la plupart des systèmes) par le signataire ou créées par l'autorité de certification. Dans ces deux cas, un système doit être utilisé lors de la création des clefs afin

d'assurer la confidentialité, l'intégrité, la disponibilité et l'utilisation légitime de celles-ci. Lors du processus de création, l'autorité de certification (si c'est elle qui crée les clefs) doit par ailleurs éviter de conserver une copie de la clef secrète de façon à ne pas devenir un détenteur non autorisé de cette clef.

g) L'autorité de time stamping

Un timbre temporel (Time stamp) est une attestation digitale cryptographiquement protégée qui prouve qu'un document existait à un moment précis. Il n'est pas difficile de prouver qu'un document existe après un autre événement. Il suffit d'inclure une référence à un événement (par exemple, un extrait d'un journal) qui s'est déroulé plus tôt, et qui n'était pas prévisible. Parfois, cela suffit à prouver qu'un document a été signé ou qu'un événement s'est déroulé après une date précise. Souvent, il est aussi important de prouver la date précise de la création d'un document. C'est beaucoup moins simple de prouver l'antériorité d'un document. On ne peut pas se baser sur la date de création ou de modifications d'un document car ces dates sont facilement falsifiables. La signature montre que c'est bien l'émetteur qui a écrit le document, mais la signature n'ajoute rien à la crédibilité de l'émetteur quant à la date de création.

La seule manière de prouver sans doute qu'un document a été écrit avant un certain moment est de créer un événement basé sur ce document, qui peut être observé par les autres. L'émetteur pourrait par exemple publier le texte de son document dans un journal. Ce qui n'est pas génial au niveau de la confidentialité. Une meilleure méthode est que l'émetteur publie une hash value de son document. Une hash value est un nombre assez grand qui est produit par une fonction de hachage dont l'argument est le texte dans son entièreté.

La fonction de hachage doit avoir trois propriétés pour servir d'empreinte à un document. La fonction de hachage doit être publique, n'importe qui peut accéder à la fonction et l'appliquer sur le texte afin de vérifier si l'empreinte correspond. La deuxième propriété d'une fonction de hachage est qu'elle est unidirectionnelle (on ne peut pas calculer les

arguments en fonction des résultats). Il faut qu'il soit impossible ou que la probabilité soit infime que deux textes différents donnent la même hash value. Soulignons que toute altération du document changera la hash value.

Il est difficile de publier chaque jour, dans un document physique, tel un journal, toutes les hash value issues des différents documents produits ce même jour. C'est pour répondre à cette difficulté qu'est apparue l'autorité de time stamping qui peut fournir un système de timbrage temporel. L'autorité de time stamping envoie un certificat qui atteste que l'émetteur lui a fait parvenir une hash value d'un document à un moment précis. Dans ce certificat, se trouvent l'heure de réception de la hash value ainsi que la hash value. En attachant, ce certificat à son document, l'utilisateur certifie le moment de création de son document. Si un quidam ne croit pas en ce certificat, il pourra vérifier la présence de la hash value auprès de l'autorité de time stamping, qui devra au préalable archiver ses certificats. Tout ce qui passe entre les mains de l'autorité de time stamping sont les hash value, donc la confidentialité du document est préservée. L'autorité de time stamping doit avoir une bonne réputation pour inspirer confiance envers ses certificats.

h) La hiérarchie des autorités de certification

Un principe de base pour avoir confiance dans un certificat est d'être sûr de l'identité correspondant au signataire du certificat. Comment être sûr de cette identité sinon par la présence d'un certificat supérieur qui en atteste. Admettons que la Belgique se munisse d'une autorité de certification par commune. Si un utilisateur de Gand désire communiquer avec un utilisateur de Charleroi, ils devront dans un premier temps s'échanger les certificats. Si le gantois n'a jamais conversé avec quelqu'un de la commune de Charleroi, il devra alors vérifier si la clef utilisée pour réaliser le certificat est bien celle qui correspond à la commune de Charleroi. Pour s'assurer de ce fait, il vérifiera le certificat de la commune de Charleroi qui est réalisé par une autorité de certification supérieur, par exemple le centre de certification régional du Hainaut. Les clefs de ce centre seront elle-même certifiées par une entité

nationale que le gantois connaît et en qui il a confiance. Nous venons donc d'introduire la notion de hiérarchie (voir figure ci-dessous) dans les certifications. Chaque clef d'une autorité de certification est elle-même certifiée par une autorité supérieure, jusqu'à l'autorité racine en qui tout le monde doit avoir confiance. On peut imaginer une autorité racine nationale, européenne ou même mondiale. Nous obtiendrons donc une chaîne de certificats, avec un certificat racine à la base de l'arbre et nous devrons faire confiance au détenteur de celui-ci.

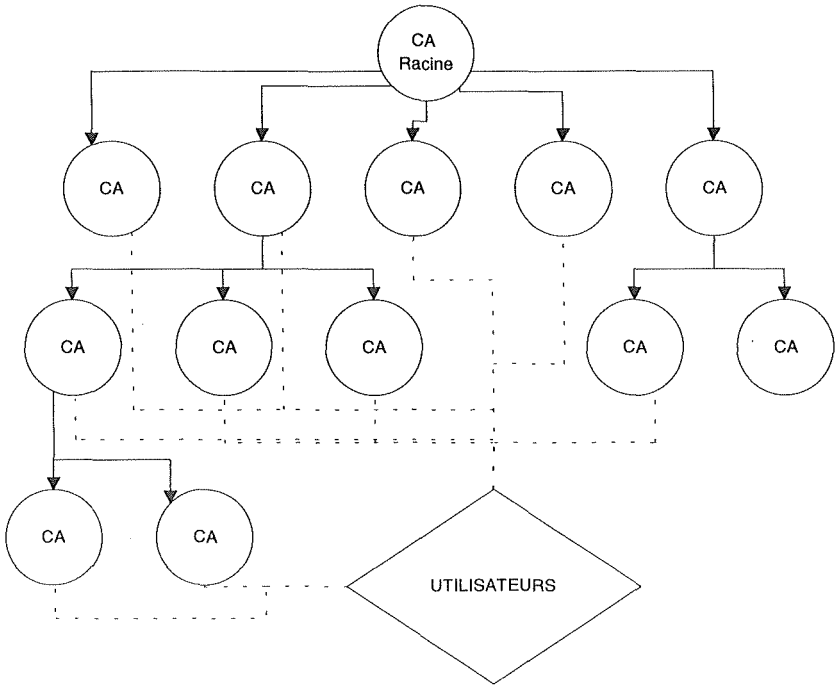

Figure 4. Hiérarchie des certifications

Nous pouvons choisir le gouvernement comme entité de confiance. Son rôle sera de certifier les clefs de certaines autres autorités de certification. La clef racine appartiendrait donc à une agence gouvernementale et les quelques autorités de certification qui répondent à des caractéristiques spécifiques (stipulées par l'état) seraient « récompensées » en voyant leurs clefs certifiées par le gouvernement. Ces autorités de certification pourraient alors, à leur tour, certifier des clefs racines d'organisations qui désirent produire et gérer leurs certificats elles-mêmes.

Plus il y a de niveaux dans l'arbre de certification, plus l'utilisateur devra

Supposons que la signature digitale de Bob est produite par CA1, dont la clef publique est certifiée par CA2, qui à son tour est certifié par CA3 qui lui est certifié par un organisme gouvernemental. Alors quelqu'un voulant s'assurer de l'identité de Bob devra vérifier quatre certificats, sauf si par le passé, il a déjà rencontré une des autorités de certification supérieure et en a déjà vérifier le certificat. L'organisme gouvernemental peut révoquer la certification de CA3, si par exemple quelqu'un a « cassé » sa clef secrète . Dans ce cas, tous les certificats qui descendent de CA3 deviendraient suspects (cette question demande à être développée vu les méthodes de time stamping qui permettrait de limiter la révocation de certains certificats). Ce schéma peut fonctionner, mais il demande à l'utilisateur beaucoup d'opérations informatiques et autant d'accès à différentes bases de données qu'il y a d'autorités de certification.

Les quelques autorités de certification qui fonctionnent actuellement (octobre 96) aux Etats-Unis, fonctionnent sans la certification d'une autorité gouvernementale. Elles certifient leurs clefs en les signant elles-mêmes sur leurs sites web. Ces certificats sont essentiellement basés sur la réputation du CA dans le monde du business. Le système de certifications lié à ce type de CA semble avoir une hiérarchie assez plate. La tendance actuelle se dirige vers ce genre de système où chaque organisation aurait sont propre CA qui ne serait certifié que par un seul autre CA. Aucun modèle de certification ne s'est encore imposé ni sur le marché américain, ni sur le marché européen.

Il existe encore une dernière possibilité de certification, c'est la cross-certification ou deux CA racines reconnaissent chacun les clefs publiques de l'autre.

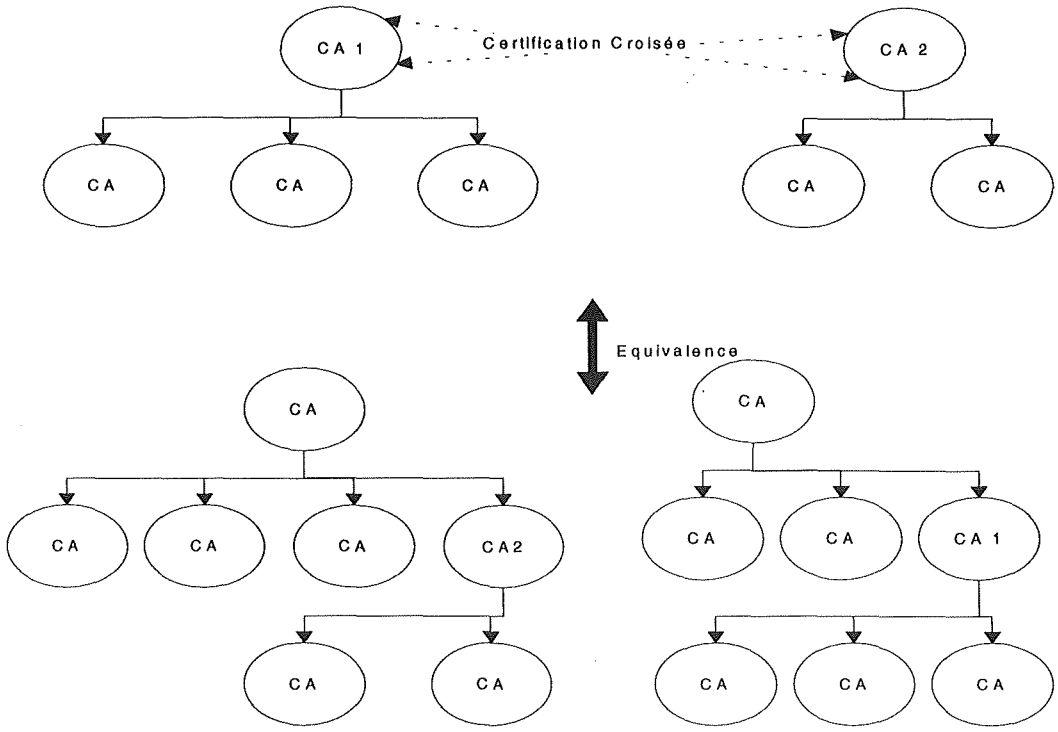

Figure 5 . schéma expliquant la certification croisée

1.5 Système de chiffrement avec dépôt de clefs (Key Escrow)

a) Introduction

Assurer le secret des communications et du stockage de données privées en utilisant la cryptographie semble la solution du futur pour les systèmes d'informations. Le but explicite de la cryptographie est de rendre difficile, voir impossible, à un tiers d'accéder aux informations sécurisées. Mais cela pose un problème (C.F. infra légalité de la confidentialité) quand le tiers est une autorité qui, pour des raisons légales ou sociales, pense avoir le droit d'accéder à l'information. Le besoin de mécanismes permettant à un tiers autorisé d'espionner, sans diminuer le secret face aux autres acteurs, se fait alors ressentir. Nous appellerons de tels systèmes les dépôts de clefs. Le dépôt de clef vise uniquement l'archivage de vecteurs permettant de retrouver des clefs secrètes.

b) Généralités

Un système de dépôt de clefs est donc un système qui permet à une personne autorisée de recouvrer les clefs secrètes d'autres personnes. Dans ce système, les clefs de chiffrement créées par les utilisateurs le sont à partir d'un vecteur générateur, qui est fourni par un dépositaire de secrets. Quand un utilisateur chiffre un message, il crée également un message de recouvrement. La personne qui désire déchiffrer un message de cette personne va demander le vecteur générateur au dépositaire de secrets et va pouvoir actionner une fonction de recouvrement, qui a comme argument le vecteur générateur et le message de recouvrement. Cette fonction a pour résultat la clef utilisée pour le déchiffrement d'un message. Il est alors possible de déchiffrer le message. Ce système est principalement applicable au chiffrement à clefs symétriques. Voyons plus en détail les différents composants d'un tel système représenté dans la figure ci-dessous.

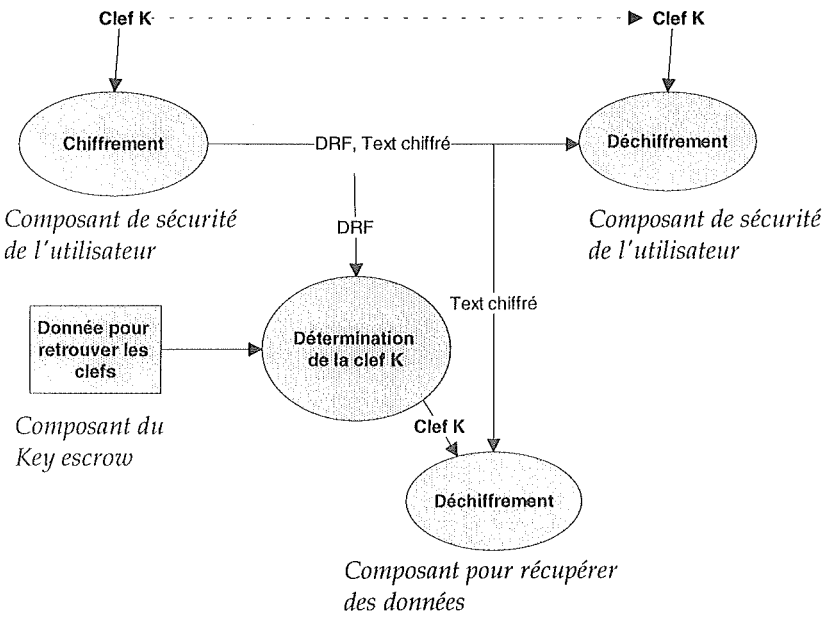

Figure 6. Schéma général d'un dépôt de clefs

([DENN96])

- Composant de sécurité de l'utilisateur : C'est un système de chiffrement hardware (Clipper Chip) ou software (AT&T Crypto Backup)

qui fournit un moyen de chiffrement capable de supporter les fonctions utiles au dépôt de clefs. Ce composant doit être capable d'attacher un data recovery field (DRF) à chaque donnée chiffrée. Ce DRF sera utilisé par le composant qui récupère les données afin de retrouver la clef de chiffrement grâce à la clef de recouvrement (donnée préservée par le composant du key escrow). Le DRF va en réalité lier la clef de chiffrement avec la clef de recouvrement. Par exemple, le DRF peut contenir la clef chiffrée par une clef de recouvrement et des informations permettant d'identifier l'agent de key escrow. Ce composant de sécurité de l'utilisateur pourra dans certains cas créer des clefs à partir de clefs de production. Soulignons que la clef de recouvrement peut être partagée entre plusieurs autorités (mécanisme du secret sharing).

- Composant du dépôt de clefs: Celui-ci est utilisé par des tiers de confiance qui gèrent le stockage, la création et l'utilisation des clefs de recouvrement pour un ensemble d'utilisateurs. Ces tiers de confiance peuvent être soit gouvernementaux, soit privés. Ils sont toujours identifiés par un nom et une localisation. Ces tiers de confiance doivent être caractérisés par un degré de fiabilité qui mesure la confiance que l'on peut avoir en eux. On doit savoir si jamais ce tiers est lui-même certifié par un autorité quelconque. Soulignons qu'il est également possible d'établir des schémas avec plusieurs tiers de confiance pour les clefs d'un seul utilisateur.

- Composant pour récupérer des données: ce composant peut, après des vérifications strictes, permettre de récupérer des données chiffrées sans disposer des clefs de déchiffrement. Celui-ci va d'abord, grâce aux données contenues dans le DRF, identifier les tiers de confiances et obtenir auprès d'eux les informations (vecteur de recouvrement) nécessaires pour retrouver la clef. Après avoir retrouvé la clef , il lui sera alors possible de déchiffrer le message.

c) Exemple d'utilisation d'un dépôt de clefs

[WALK96],[MAHE96]

Le projet du gouvernement américain visant à instaurer un système de dépôt de clefs général basé sur un chip appelé Clipper a été assez fortement rejeté par la population d'utilisateurs pour plusieurs raisons. Les premières raisons évoquées sont relatives à la protection de la vie privée si chère aux Etats-Unis. Mais la raison principale est le coût hardware de remplacement du matériel dans le cas ou certains secrets seraient rendus publiques. De plus, ce système de dépôt de clefs ne rend aucun service supplémentaire à l'utilisateur. C'est pour cela que des développements ont été apportés pour réaliser des dépôts de clefs utiles pour l'utilisateur et répondant aux désirs des gouvernements. Décrivons maintenant l'utilisation d'un tel système appelé CryptoBackup, qui est un système mis au point par AT&T.

Ce système a pour but de répondre à certaines attentes de ses utilisateurs comme de permettre de déchiffrer un ancien document malgré la perte de la clef correspondante ou de déchiffrer un document si le possesseur de la clef est absent. Le principe est similaire à l'utilisation de clefs physiques. Si un membre du personnel part en vacance un double de la clef de son bureau est laissée à quelqu'un en qui il a confiance. Décrivons un exemple complet de ce cryptosystème.

•Alice choisit une personne (Bob) en qui elle a confiance et qui est reconnue par son organisation. Elle pourrait également choisir plusieurs personnes.

A⇒B: demande

•Bob lui fournit une clef de production (MKV masterkey vector) qui lui permettra de créer des clefs. Bob a également signé cette clef.

$B \Rightarrow A: S_{SK\,b}(MKV)$

•Alice, après avoir vérifié la signature, installe la clef de production dans son générateur de clefs. Elle peut également utiliser un générateur qui produit des clefs à partir de plusieurs MKV.

•Chaque fois qu'Alice crée un nouveau document chiffré, elle lui associe

un « Backup recovery vector » qui permet à Bob de retrouver la clef utilisée par Alice.

A: BRV,E_K(message)

•On peut supposer que B a créé MKV avec une fonction à sens unique F(V)=MKV et que K=RECOU(V,BRV). Si Alice a décidé de faire confiance à n tiers alors K= RECOU(V_i ,BRV) où i =1,.....m et m<=n

Grâce à un tel schéma, on répond aux attentes du gouvernement. En effet, il suffit au gouvernement de contacter Bob pour pouvoir espionner Alice. L'utilisation de ce système permet à un utilisateur de pouvoir récupérer des clefs perdues. D'autres exemples d'utilisations et une description plus technique se trouvent dans[MAHE96]

1.6 La carte à puce

Nous venons de parler de moyens de recouvrir des clefs secrètes perdues ou non disponibles mais il reste une question importante: comment préserver les clefs secrètes dans un réseau ouvert. Cette question est fondamentale, car un utilisateur qui ne peut préserver ses clefs secrètes est soumis au risque de voir une tierce personne malveillante, disposant de sa clef, se faire passer pour lui. A première vue, il est possible de préserver ces clefs sur des partitions de disques durs. Mais ce système n'est pas très sûr. La plupart des systèmes fonctionnant sous UNIX sont régulièrement craqués et se sont eux les plus répandus. Il faut alors se retourner vers un autre moyen pour préserver le secret. Il semble qu'un moyen de plus en plus utilisé soit la carte à puce.

Les cartes à mémoire intégrée et les cartes à microprocesseur entrent dans les techniques utilisées pour la sécurisation de la clef privée. La carte à mémoire intégrée peut ainsi être conçue afin de conserver dans sa mémoire la clef privée, un PIN (personnal identification number) et facultativement la clef publique de l'usagé ainsi que d'autres informations. Les informations ainsi accessibles peuvent alors être traitées par un ordinateur qui exécute les tâches de signature, de chiffrement et de déchiffrement. La carte à

microprocesseur peut quant à elle contenir les mêmes informations. Elle offre cependant une plus grande flexibilité puisque les opérations de signature, de chiffrement et de déchiffrement peuvent être réalisées par l'entremise du microprocesseur contenu dans la carte. En effet, « seules les cartes à microprocesseur contiennent, en plus des circuits intégrés, un processeur et des logiciels qui les rendent capables d'effectuer des opérations internes pouvant servir à la vérification ou au chiffrement de données ». Les cartes à puce peuvent également être associées à des mécanismes biométriques de contrôle d'accès sophistiqué comme l'empreinte palmaire.

Avant de commencer à détailler les composants des cartes à puce, rappelons que les cartes à puce sont des cartes contenant des circuits intégrés. Les cartes magnétiques et les cartes optiques ne sont pas reprises sous cette appellation. Les cartes à puce doivent répondre à certaines normes ISO notamment à propos de la position des contacts et du voltage utilisé.

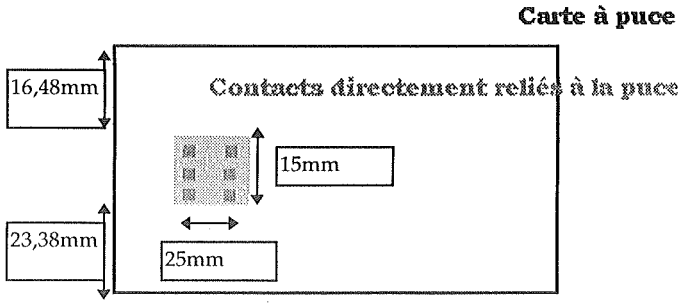

Figure 7. Situation de la puce sur une carte à puce

a) Les cartes à mémoire

[ZORE94]

Comme son nom l'indique, une carte à mémoire simple renferme seulement une certaine quantité de mémoire, sans aucune protection particulière. Cela signifie que n'importe qui peut y lire ou y écrire des informations à l'aide d'un équipement approprié librement disponible dans le commerce ou même souvent très simple à construire par ses propres moyens.

La plupart des cartes à mémoire sont réalisées en technologie EEPROM (Electrically Erasable Programmable ROM)., et sont donc réutilisables. Leur capacité est de l'ordre de quelques kilobits. Le fait que ces mémoires soient non volatiles mais que l'on puisse quand même les réutiliser, les rend particulièrement attractives pour y stocker la clef privée ou secrète.

Parlons des différentes types de mémoires non volatiles utilisées dans les cartes a puces. L'information permanente peut être stockée dans la ROM (Read Only Memory). Dans ce cas, toute l'information est enregistrée au moment de la création du chip ou dans une étape postérieure de programmation. Dans le premier cas, l'information est incluse dans le masque photolithographique qui est utilisé pour produire le chip. C'est de la mémoire ROM pure. Le problème de ces mémoires, qui sont essentiellement hardware, c'est l'irréversibilité de l'écriture. Il est impossible de corriger des erreurs de programmation. Mais il existe d'autres types de mémoire ROM, qui sont des mémoires suivant le principe de write once, read-many. Ces ROM programmables (PROM) permettent à l'utilisateur de programmer le chip en brûlant certains fusibles de celui-ci. Dans ce cas la mémoire est également permanente. Une fois l'information écrite, elle reste à jamais. La flexibilité s'améliore avec l'EPROM (Erasable PROM). Dans ce cas, la PROM peut être effacée par une action extérieure, habituellement une exposition aux rayons ultraviolets. Cette technologie ne s'adapte pas aux cartes à puce dont le circuit est encapsulé. Et enfin nous arrivons au dernier type de mémoire non volatile: les EEPROM. Une technologie qui fournit des mémoires non volatiles qui sont effaçables par un simple signal électrique. Cette mémoire est un candidat parfait pour les utilisations des cartes à mémoire personnalisées. Il est facile d'y inscrire des informations telles que des clefs ou des noms. Un problème mineur des EEPROM est que les opérations d'effacement et d'écriture demandent un voltage supérieur à celui des opérations de lecture. La norme ISO a tenu compte de ce problème

Pour mériter l'appellation de « carte à mémoire personnalisée », une carte à puce doit contenir au moins un des trois systèmes de protection suivants, réalisés en logique câblée sans le recours au microprocesseur:

•Zone protégée en écriture après destruction d'un fusible.

•Zone protégée en lecture et écriture par un « code porteur »(PIN), le porteur étant défini comme l'utilisateur de la carte, avec blocage de la carte au bout de quatre présentations d'un PIN erroné

•Protection par un « code émetteur »(l'émetteur étant l'organisme qui délivre la carte).

Les capacités disponibles sont généralement inférieures à quelques kilobits. Cette capacité permet juste le stockage des clefs, L'évolution de ces cartes permet des applications avancées comme les portefeuilles électroniques.

b) Les cartes à microprocesseur

La carte à microprocesseur (dite aussi microcalculateur) est le nec plus ultra en matière de cartes à puce: C'est un véritable micro-ordinateur qui rassemble une unité centrale, des mémoires de programme et des mémoires de données spécialement agencées.

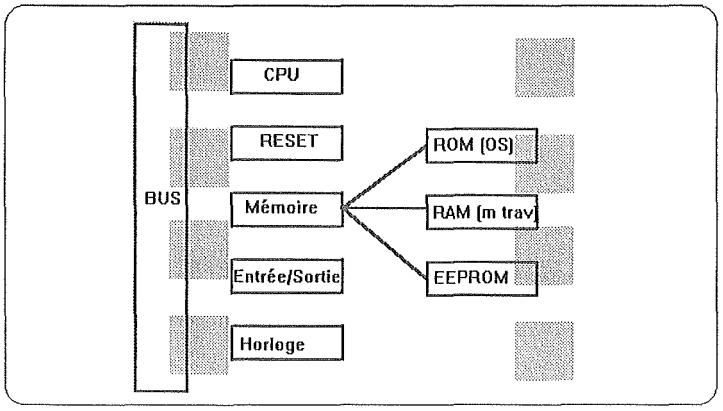

Figure 8. Les différents composants d'une carte à microprocesseur

Les cartes à microprocesseur utilisent des mémoires volatiles (RAM). Même si la carte est sous tension, ces mémoires ne préservent pas les informations qu'elle possède très longtemps. Il y a deux sortes de mémoire RAM :les statiques (SRAM) qui n'ont pas besoin d'être rafraîchies et les dynamiques (DRAM) qui sont rafraîchies régulièrement mais qui sont pratiques pour être utilisées comme mémoire de travail. De plus, un autre

avantage des SRAM est que leur prix est bas par rapport au DRAM.

Les mémoires non volatiles sont également utilisées pour stocker le système d'exploitation (PROM) ou le programme (EEPROM).

Les CPU utilisés à l'heure actuelle emploient en général des instructions de 8 bits ou 16 bits (taille du bus) à une fréquence de 5 Mhz à 14 Mhz. D'une vue extérieure le comportement d'une smart card est essentiellement dépendant de son SCOS (Smart Card Operating System). Le développement des cartes à puce est très important pour l'instant, la firme GEMPLUS parle notamment de processeur Risk à 64 bits.

Une carte à microprocesseur possède également d'autres circuits. Comme le circuit Reset qui, quand il est activé, remet le système au point de départ. Au Reset, l'ordinateur se réinitialise et charge son SCOS avant de lancer les programmes en mémoire . Le circuit de l'horloge reçoit un signal extérieur pour permettre la synchronisation. Le fait que l'horloge soit issue de l'extérieur limite les possibilités de contrôle temporel de la carte. La dernière sorte de circuits existants sont des circuits de communication. Ceux-ci sont souvent mis à l'épreuve par les contacts avec l'interface, notamment à cause de la saleté sur les contacts.

Produit informatique à part entière, la carte à microprocesseur contient plusieurs systèmes de protection :

- Zone protégée en écriture ou en écriture et lecture par un code secret émetteur.
- Zone protégée en écriture et lecture par un PIN.
- Blocage de la carte après présentation de codes secrets erronés, mais avec possibilité de réhabilitation par l'organisme émetteur.
- Mise en oeuvre d'algorithmes cryptographiques pour assurer les transferts de données.

Dans ce travail, l'utilité essentielle de la carte à microprocesseur est que l'on peut l'utiliser pour le chiffrement et le déchiffrement . Un article de De Waleffe et Quisquater donne une méthode pour appliquer rapidement l'algorithme RSA [WALE90]. Cette carte à puce semble un très bon moyen pour préserver les secrets.

1.7 Conclusion

Après ce premier chapître, les notions de clefs asymétriques nous ont permis de comprendre le fonctionnement du chiffrement et de la signature électronique. Nous avons pu saisir alors l'utilité des tiers de confiance et leurs différents rôles. Nous avons aussi expliqué le principe du dépôt de clefs et des cartes à puce. Ces deux technologies sont utilisées l'une pour recouvrir des clefs secrètes et l'autre pour les préserver. Nous pouvons, maintenant passer à une étude plus systématique des protocoles de communication impliquant des tiers de confiance.

PROTOCOLES DE DISTRIBUTION DES CLEFS ET TIÈRS DE CONFIANCE

2. Protocoles de distribution des clefs et tiers de confiance

2.1 Introduction

Le but de ce chapître est de faire le tour des protocoles utilisés avec des systèmes à chiffrement asymétrique. Le premier protocole est relativement simple : chacun des utilisateurs va chercher la clef publique de l'autre auprès d'un tiers de confiance qui ici ne tient que le rôle de distributeur de clefs. On ne sait absolument pas comment ce tiers de confiance dispose des clefs.

Dans le deuxième protocole, nous remarquons l'apparition des certificats qui permettent d'éviter des connexions des deux parties au tiers de confiance. Le tiers de confiance est devenu une autorité de certification et un distributeur de certificats.

Dans ces deux premiers protocoles, il n'est pas question de la génération des clefs qui peut être réalisée par les tiers de confiance ou par les utilisateurs eux-mêmes. Nous ne rencontrons donc pas encore le rôle de générateur de clef.

Dans le troisième protocole, le tiers de confiance a la charge de générer une clef de session qui n'est valable que pour une session de communication. Le rôle de générateur de paires de clefs est laissé à l'utilisateur. Mais les procédures d'admission ne sont pas précisées. Le rôle d'enregistrement n'apparaît pas encore.

Dans le dernier protocole, de petites modifications ont été apportées pour séparer le rôle d'enregistrement des autres rôles du tiers de confiance. Nous avons donc l'apparition d'un deuxième tiers de confiance qui est une autorité d'enregistrement. Ce protocole permet également la mise au point d'un système permettant la non répudiation. De plus ces deux derniers protocoles permettent le dépôt de clefs.

Petit rappel sur les notations utilisées

Ces notations ont été choisies en vue de respecter les normes utilisées dans le cours de « fiabilité et sécurité des systèmes informatiques ».

sk_A:= clef secrète de A (secret key)

pk_A:= clef publique de A (public key)

$Epk_A(P)$:chiffrement de M avec la clef publique de A (chiffrement)

$Dsk_A(C)$:déchiffrement de C avec la clef secrète de A (déchiffrement)

$Ssk_A(P)$:signature de M avec la clef secrète de A (signature)[= $Dsk_A(M)$]

$Vpk_A(C)$:authentification de C avec la clef secrète de A (Vérification)[= $Esk_A(C)$]

Etk(P):chiffrement de M avec une clef de session (Chiffrement)

2.2 Protocole de communication simple

REF:NEED 78

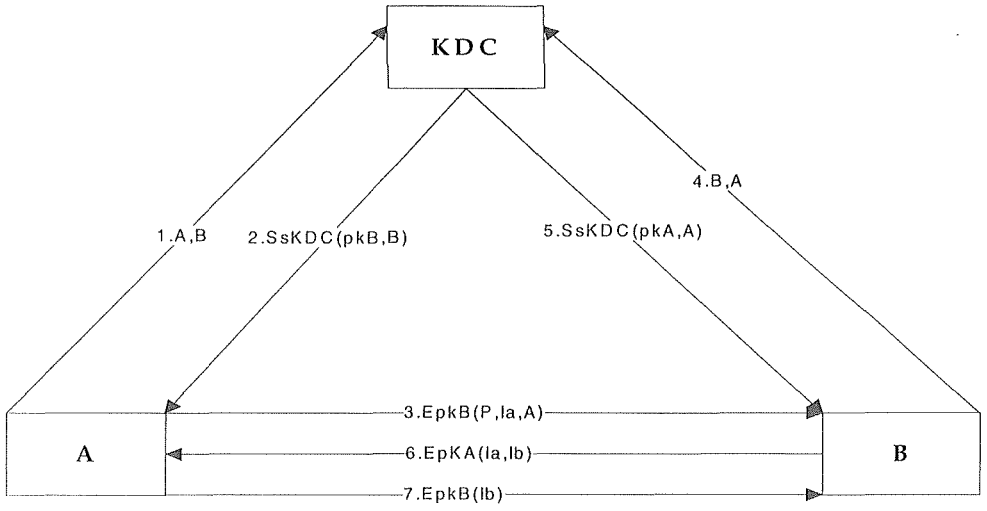

Figure 9. Protocole de communication simple

a) Prérequis au début du protocole

Il existe trois entités qui dialoguent dans un réseau ouvert. Un système ouvert est défini [DAVI97] comme un système dans lequel aucune entité administrative ou légale ne contrôle les activités de communication, le stockage d'informations ou les utilisateurs. Chacune des entités connaît sa

paire de clefs asymétriques et la clef publique de l'unité appelée KDC (keys distribution center). Celle-ci connaît, en plus, l'ensemble des clefs publiques de tous ses « adhérents ».

Toutes les entités connaissent l'algorithme de chiffrement asymétrique commun(ex RSA, El Gamal). On peut supposer que ces informations ont été placées sur un support distribué aux adhérents et au KDC de manière sécurisée.

A: sk_A , pk_A , pk_{KDC}

B: sk_b , pk_b , pk_{KDC}

KDC: sk_{KDC} , pk_X (représente ici toutes les clefs publiques appartenant aux utilisateurs affiliés au centre distributeur de clefs)

b) Protocole de communication

Etape 1 :Demande de la clef publique du correspondant

A⇒KDC: A, B
KDC: pk_B

A envoie au centre de distribution son identité et l'identité de son futur correspondant.

KDC va retrouver dans ses tables la clef publique pk_B appartenant au correspondant du demandeur.

Etape 2 :Envoi de la clef signée

KDC⇒A: $Ssk_{KDC}(pk_B , B)$
A: $Vpk_{KDC}(Ssk_{KDC}(pk_B , B))= pk_B , B$

KDC chiffre avec sa clef privée la clef publique de B ainsi que son identité et envoie l'information à A.

A déchiffre le message et connaît pk_B qui est authentifiée par KDC. Il est donc sûr que c'est bien la clef publique de son correspondant qu'il a reçue. Il doit pour cela avoir confiance dans KDC. Nous remarquons ici , l'apparition

de la notion de certificat dans le protocole, car le KDC certifie le lien entre la clef et l'identité.

Dans cette étape, on remarquera l'apparition des rôles de certification et de distributions des certificats.

Etape 3 :Envoi du message

$A \Rightarrow B$: $Epk_B(P, I_A, A)$
B: $Dsk_B(Epk_B(P, I_A, A)) = P, I_A, A$

A envoie à B un message, un nonce et son identité chiffré avec la clef publique de B . Seul B est capable de lire ces informations. Ceci étant certifié par l'authentification de la clef à l'étape 2.

B sait qu'il est le seul a avoir pu déchiffrer le message mais il n'a aucune preuve de l'identité de l'envoyeur.

Etape 4 Demande de la clef publique du correspondant

$B \Rightarrow$KDC: B, A
KDC: pk_A

Cette étape est similaire à la première étape.

B envoie au centre de distribution son identité et l'identité de son correspondant actuel qui n'est pas encore authentifiée.

KDC va retrouver dans ses tables la clef publique pk_A

Etape 5 : Envoi de la clef signée

$KDC \Rightarrow B$: $Ssk_{KDC}(pk_A, A)$
B: $Vpk_{KDC}(Ssk_{KDC}(pk_A, A)) = pk_A, A$

KDC chiffre avec sa clef privée la clef publique de A ainsi que son identité et envoie l'information à B.

B déchiffre le message et connaît alors pk_A qui est authentifiée par KDC. La question de confiance dans le KDC est toujours en vigueur.

Etape 6 :Envoi d'un accusé de réception de P

B⇒A: $Epk_A(I_A, I_B)$
A: $Dsk_A (Epk_A (I_A , I_B))= I_A , I_B$

B envoie à A deux nonces: celui reçu de lui même et un nouveau en vue d'une authentification.

A est sûr que B a reçu le message et a pu le déchiffrer.

Etape 7 :Confirmation de l'identité de l'émetteur

A⇒B: $Epk_B(I_B)$
B: $Dsk_B (Epk_B (I_B))= I_B$

A envoie à B le dernier nonce reçu de B en vue d'une authentification.

B est sûr que c'est bien A qui a envoyé le message de départ.

c) Postcondition

A: $sk_A , pk_A , pk_B , pk_{KDC}$

B: $sk_B , pk_B , pk_A , pk_{KDC}$

KDC: sk_{KDC} , pk_X

Après le protocole, les deux parties connaissent les clefs publiques de leur partenaires, si elles comptent reprendre une communication avec celui-ci, alors elles peuvent garder les clefs et elles n'ont plus besoin du distributeur de clefs. Sauf évidemment si elles veulent s'assurer que les clefs sont toujours valides.

d) Commentaires

Un premier problème de ce protocole est le manque d'explications à propos de l'acceptation de la clef publique par le centre distributeur des clefs. De ce même point de vue, on ne sait pas comment l'utilisateur obtient la clef publique du centre de distribution des clefs, ni comment il est sûr que celle-ci est bien certifiée.

Dans ce protocole, on ne considère pas le problème de la non-

répudiation de l'origine. La non-répudiation de l'origine est le fait que le destinataire peut garder une preuve certaine que le document reçu appartient bien à l'émetteur. Le document reçu est simplement le chiffrement du message avec la clef publique du destinataire et la preuve de l'identité de l'émetteur est dynamique, on ne peut donc pas garder la preuve formelle de l'identification. La non-répudiation serait éventuellement possible si les messages étaient signés et si le KDC gérait les clefs. Quand nous parlons de régir la gestion des clefs, nous nous référons à la fonction d'archivage qui est imputée à l'autorité de distribution des clefs. L'archivage doit comprendre les dates de validité des clefs ainsi que les révocations et les suspensions. Car une vérification en cas de litige est toujours possible.

Il est possible d'étendre ce protocole de manière à ce qu'il y ait plusieurs échanges d'informations. Il suffit pour cela que l'étape 6 soit remplacée par $Ekp_A(I_A, I_B, \text{Réponse})$, de même pour les étapes suivantes jusqu'à la fin de la session de communication. La répudiation des clefs qui fait partie du rôle de distribution des certificats n'est pas décrite ici.

2.3 Protocole de communication à clef publique directement certifiée (x.509)

[CART95] ,[LAUN90] ,[SCHN93]

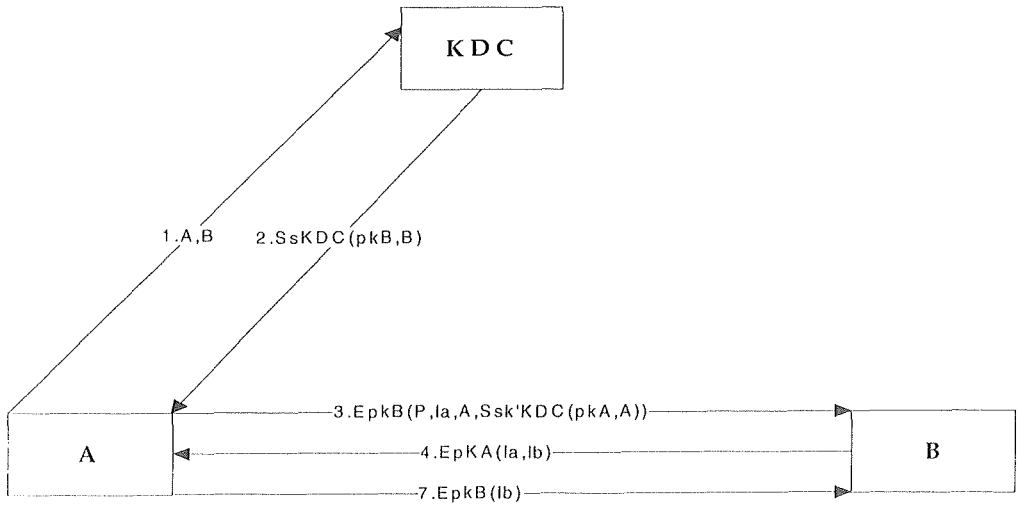

Figure 10. Protocole de communication directement certifiée

a) Prérequis au début du protocole

Dans ce protocole nous avons toujours trois unités qui dialoguent dans un réseau ouvert. Chacune connaît sa clef privée et la clef publique du centre de distribution des clefs. Celle-ci connaît, en plus, l'ensemble des clefs publiques de tous ses « adhérents ».

Elles connaissent également une information qui leur est particulière et reconnaissable par tout le monde. Cette information est en réalité un certificat.

A: sk_A , pk_A , pk_{KDC}, $Ssk_{KDC}(pk_A, A)$

B: sk_b , pk_b , pk_{KDC} , $Ssk_{KDC}(pk_B, B)$

KDC: sk_{KDC} , pk_X

b) Protocole de communication

Etape 1 : Demande d'information sur A

A⇒KDC: A,B
KDC: pk_B

A envoie au centre de distribution son identité et l'identité de son futur correspondant

KDC va retrouver dans ses tables la clef publique pk_B.

Etape 2 : Réponse à la demande

KDC⇒A:$Ssk_{KDC}(pk_B , B)$
A: $Vpk_{KDC}(Ssk_{KDC}(pk_B , B))= pk_B , B$

KDC chiffre avec sa clef privée la clef publique de B ainsi que son identité et envoie l'information à A.

A déchiffre le message et connaît pk_B qui est authentifiée par KDC. Dans le protocole x509 qui est très semblable à celui décrit ici, le centre de distributeur des clefs fournit un certificat qui comprend la clef publique, l'identité, un chemin de certification entre A et B et les algorithmes de

chiffrement utilisés. Le chemin de certification permettra à B de vérifier s'il le désire la validité du certificat.

Etape 3 : Envoi du message

$A \Rightarrow B$: $Epk_B(M, I_A, A, Ssk_{KDC}(pk_A, A))$
B: $Dsk_B(Epk_B(M, I_A, A)) = M, I_A, Ssk_{KDC}(pk_A, A), A$
B: $Vpk_{KDC}(Ssk_{KDC}(pk_A, A)) = pk_A, A$

A envoie à B un message, sa clef publique authentifiée par KDC et son identité chiffrée avec la clef publique de B . Seul B est capable de lire ces informations. Ce dernier élément de l'envoi est la nouveauté de ce protocole par rapport au précédent.

B sait qu'il est le seul à savoir déchiffrer ces données.

B déchiffre ou du moins vérifie l'authenticité du certificat et connaît pk_A qui est authentifiée par KDC. Il est donc sûr que seul A saura déchiffrer un message chiffré avec pk_A. Il est aussi sûr de l'identité de A.

Etape 4 : Envoi de l'accusé de reception

$B \Rightarrow A$: $Epk_A(I_A, I_B)$
A: $Dsk_A(Epk_A(I_A, I_B)) = I_A, I_B$

B envoie à A le nonce reçu de lui même pour confirmer la réception du message.

A est sûr que B a reçu le message.

Etape 5 Confirmation de l'identité de l'envoyeur

$A \Rightarrow B$: $Epk_B(I_B)$
B: $Dsk_B(Epk_B(I_B)) = I_B$

A envoie à B le dernier nonce reçu de B en vue d'une authentification supplémentaire.

B est sûr que c'est bien A qui a envoyé le message de départ.

c) Postcondition

A: sk_A , pk_A , pk_B , pk_{KDC}, $Ssk_{KDC}(pk_A)$

B: sk_B , pk_B , pk_A , pk_{KDC}, $Ssk_{KDC}(pk_B)$

KDC: sk_{KDC} , pk_X

Après le protocole, les deux parties connaissent la clef publique de leur partenaire. Si elles comptent reprendre une communication avec celui-ci, alors elles peuvent garder les clefs et elles n'ont plus besoin du distributeur de clefs.

d) Commentaires

Ce protocole est un protocole qui est très utilisé dans le monde actuel sous le nom de protocole X.509 (du nom du certificat auquel il est associé). Dans l'utilisation de ce protocole, l'utilisation de plusieurs centres de certification est possible mais alors les clefs qui certifient doivent également être certifiées. Donc une nouvelle notion apparaît : les chemins de certification qui reprennent tous les certificats vers une autorité racine.

Grâce à l'utilisation du certificat x.509, les deux parties peuvent utiliser des algorithmes de chiffrement différents si ceux-ci sont standards et connus par tous.

Ce protocole ne permet toujours pas la non-répudiation de l'origine et l'identification est toujours dynamique. Pour permettre la non-répudiation il suffirait de signer les messages avec $S_{SA}(M, B)$. Mais ne pensons pas que nous pouvons nous passer alors des nonces car ils ont leur utilité en cas de rejeu (renvoi d'un même message deux fois).

Nous pouvons remarquer également l'importance de l'aspect aléatoire de ces nonces. Car, si la machine qui crée la série de nombres aléatoires est prévisible, la validité de ce protocole est à remettre en cause. Si une personne mal intentionnée (C) désire se faire passer pour A auprès de B, elle va obtenir un certificat C_A et un certificat C_B en deux demandes de connexion auprès de KDC. Ensuite, elle va exécuter quelques conversations avec B pour connaître le début de la séquence des nombres aléatoires. De cette séquence, elle pourra dériver la suite des nonces et se faire passer pour A dans le schéma

qui se trouve dans la figure ci-dessous.

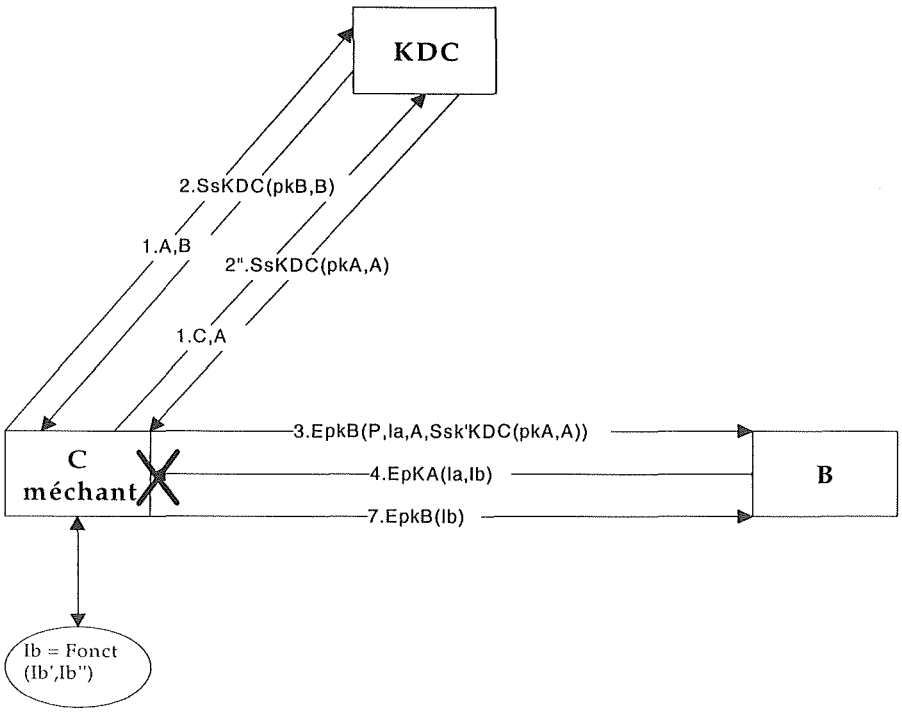

Figure 11. Exemple de fausse identification

2.4 Protocole de communication à clef publique et clef de session

[RAMA92]

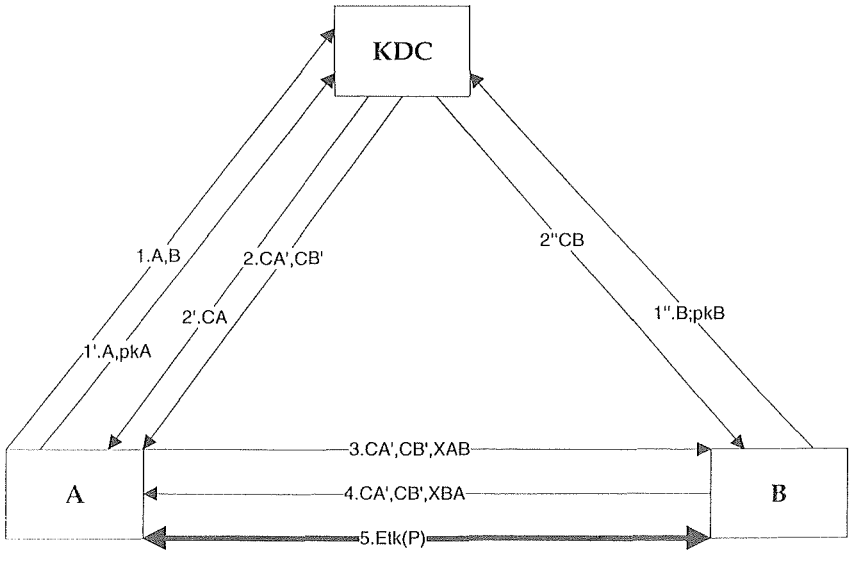

Figure 12. Protocole de communication à clef publique et clef de session

a) Prérequis au début du protocole

Il existe encore trois unités qui dialoguent dans un réseau ouvert. Chacune peut générer sa clef privée et connaît la clef publique de l'unité appelée KDC (keys distribution center).

Toutes les unités connaissent l'algorithme de chiffrement asymétrique commun ainsi qu'un algorithme de chiffrement symétrique.

A: GEN=(sk_A , pk_A), pk_{KDC}

B : GEN=(sk_B , pk_B), pk_{KDC}

KDC: sk_{KDC}

b) Protocole de communication

Etape 1 : Enregistrement des parties

A: GEN= sk_A , pk_A

A⇒KDC : A , pk_A

KDC : STORE pk_A

KDC⇒A:$Ssk_{KDC}(A, pk_A , T_{0A})= C_A$

A: $Vpk_{KDC}(C_A)=A, pk_A , T_{0A}$

A génère sa clef privée et sa clef publique.

A adhère au tiers certificateur en envoyant son identité et sa clef publique.

KDC va stocker la clef publique de A dans ses tables.

KDC envoie à A un certificat qui est composé de son identité, sa clef publique et un « time stamp », le tout codé avec la clef privée de KDC. A vérifie que le certificat est correct. Ici KDC joue les rôles d'autorité de certification et d'autorité de distribution des certificats.

B: GEN= sk_B , pk_B

B⇒KDC : B , pk_B

KDC : STORE pk_B

KDC⇒B : $Ssk_{KDC}(B, pk_B , T_{0B})=C_B$

B: $Vpk_{KDC}(C_B)=B, pk_B , T_{0B}$

B génère sa clef privée et sa clef publique. B adhère au tiers certificateur en envoyant son identité et sa clef publique.

KDC va stocker la clef publique de B dans ses tables.

KDC envoie à B un certificat qui est composé de son identité, sa clef publique et un « time stamp », le tout codé avec la clef privée de KDC.

Etape 2 : Connexion

A⇒KDC: A,B

KDC⇒A :$Ssk_{KDC}(A, pk_A ,Epk_A(tk) , T_1), Ssk_{KDC}(B, pk_B ,Epk_B(tk) , T_1)= C_A ,C_B$

A: $Vpk_{KDC}(Ssk_{KDC}(A, pk_A ,Epk_A(tk) , T_1))= A, pk_A ,Epk_A(tk) , T_1$

A: $Dsk_A(Epk_A(tk))= tk$

A: $Vpk_{KDC}(Ssk_{KDC}(B, pk_B ,Epk_b(tk) , T_1))= B, pk_b ,Epk_b(tk) , T_1$

A demande au KDC l'adresse de B et une clef de session

A: $Epk_B(Ssk_A (tk, T_2))= X_{AB}$

A⇒B :(C_A ,C_B ,X_{AB})

B: $Epk_A(Ssk_B (tk, T_3))= X_{BA}$

B⇒A: :(C_A ,C_B ,X_{BA})

KDC retrouve dans ses tables les informations recherchées et les signes avant de les envoyer à A.

A authentifie l'envoi de KDC et déchiffre la clef de session ainsi que les informations demandées.

A signe et chiffre la clef de session pour que B soit sûr de son correspondant.

C-A est un certificat particulier car en plus de l'authentification de A et de pk_A. Il permet de distribuer une clef de session.

Etape 3 : Communication

A⇔B : Etk(M)

Etape 4 : Fin de session

Destruction de tk

c) Postcondition

A: sk_A , pk_A , pk_B , pk_{KDC}

B: sk_B , pk_B , pk_A , pk_{KDC}

KDC: sk_{KDC} , pk_X

Après le protocole, les deux parties connaissent la clef publique de leur partenaire.

d) Commentaires

Dans la première étape, nous soulevons un problème central des tiers de confiance. Comment le KDC peut-il être sûr de l'identité des personnes qui s'enregistrent dans le système? Car nous pouvons créer nous-mêmes nos clefs donc n'importe qui peut le faire et se faire passer pour nous. Une solution à ce problème est de faire réaliser les clefs par l'autorité de confiance qui va authentifier la clef et l'identité du propriétaire avant de les enregistrer sur un support (smart card ou autre). L'utilisateur pourra alors s'affilier à un réseau TTP . On peut imaginer une autre solution où l'utilisateur crée ses propres clefs, les place sur un support et va physiquement auprès du centre d'enregistrement des clefs pour faire certifier sa clef publique.

On remarque en réalité qu'il peut y avoir une séparation du rôle d'enregistrement des autres rôles du tiers de confiance.

Un avantage de ce protocole est la possibilité de l'intégration d'un dépôt de clefs chez le centre distributeur de clefs, qui pourrait stocker des clefs utilisées pour produire les clefs de session pour chaque entité et adjoindre un champ de recouvrement à ses certificats.

Dans ce protocole remarquons l'utilisation explicite de certificats et le fait que la remarque sur la non répudiation est toujours de vigueur.

2.5 Protocole de communication 'amélioré'

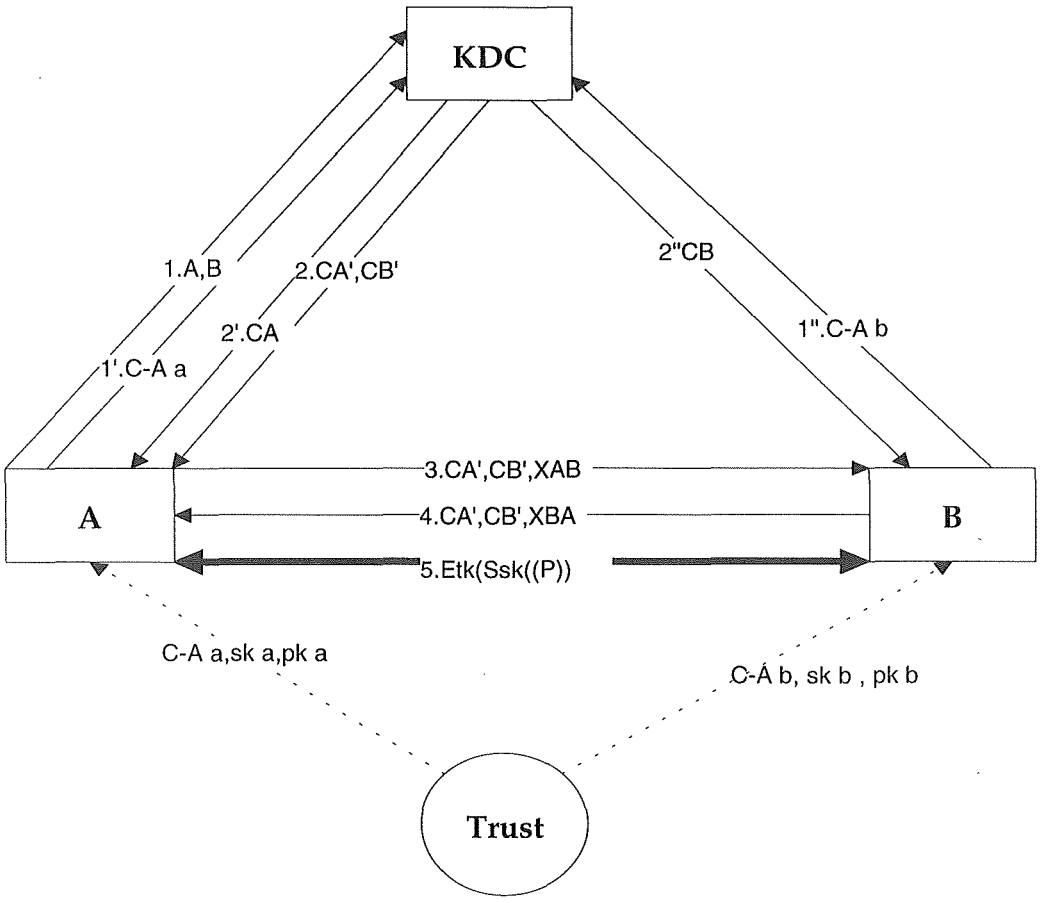

Figure 13. Protocole de communication amélioré

Dans ce protocole nous avons essayé de garder les éléments intéressants des protocoles précédents en séparant le rôle d'enregistrement, de ceux de distribution et de certification, tout en permettant la non-répudiation de l'origine.

a) Prérequis au début du protocole

Il existe quatre unités dont trois qui dialoguent dans un réseau ouvert. Chacune connaît la clef publique de l'unité appelée KDC.

Toutes les unités connaissent l'algorithme de chiffrement asymétrique commun ainsi qu'un algorithme de chiffrement symétrique.

A: pk_{KDC}

B : pk_{KDC}

KDC: sk_{KDC} ,pk_{trust}

Etape 1 : obtention des clefs

A obtient par un réseau sécurisé un support sur lequel est placé ses différentes clefs et un certificat d'admission.

sk_A, pk_A, $Ssk_{trust}(A, pk_A)$, A
$Ssk_{trust}(A, pk_A)$, A :=C-A_A (certificat d'admission).

A peut générer lui-même ses clefs et faire valider, auprès du centre d'enregistrement, le lien entre sa clef publique et son identité ainsi que faire vérifier sa possession de clef privée correspondante. Idem pour B .

Etape 2 : Enregistrement des parties

A⇒KDC : C-A_A
KDC : $Vpk_{trust}(Ssk_{trust}(A, pk_A))=A, pk_A$
KDC : STORE pk_A
KDC⇒A:$Ssk_{KDC}(A, pk_A , T_{0A})=C_A$
A: $Vpk_{KDC}(Ssk_{KDC}(A, pk_A , T_{0A}))=A, pk_A , T_{0A}$

A adhère au tiers certificateur en envoyant son identité et sa clef publique qui ont été préalablement authentifiées.

KDC va stocker la clef privée de A dans ses tables.

KDC envoie à A un certificat qui est composé de son identité, sa clef publique et un « time stamp », le tout codé avec la clef privée de KDC.

Idem pour B.

Etape 3 : connexion

A⇒KDC: A,B
KDC⇒A :$Ssk_{KDC}(A, pk_A ,Epk_A(tk) , T_1), Ssk_{KDC}(B, pk_B ,Epk_B(tk) , T_1)= C_A$,C_B
A: $Vpk_{KDC}(Ssk_{KDC}(A, pk_A ,Epk_A(tk) , T_1))= A, pk_A ,Epk_A(tk) , T_1$
A: $Dsk_A(Epk_A(tk))= tk$

A: $Vp_{kKDC}(Ssk_{KDC}(B, pk_B, Epk_b(tk), T_1)) = B, pk_b, Epk_b(tk), T_1$

A demande au KDC l'adresse de B et une clef de session

A: $Epk_B(Ssk_A(tk, T_2)) = X_{AB}$

A⇒B :(C_A ,C_B ,X_{AB})

B: $Epk_A(Ssk_B(tk, T_3)) = X_{BA}$

B⇒A: :(C_A ,C_B ,X_{BA})

KDC retrouve dans ses tables les informations recherchées et les signe avant de les envoyer à A.

A authentifie l'envoi de KDC et déchiffre la clef de session ainsi que les informations demandées.

A signe et chiffre la clef de session pour que B soit sûr de son correspondant.

Etape 4 : communication

A⇒ B : $Etk(Ssk_A(M))$

B⇒ A : $Etk(Ssk_B(M))$

Etape 5 : fin de session

Destruction de tk

b) Postcondition

A: sk_A , pk_A , pk_B , pk_{KDC},C-A_A, C_A

B: sk_B , pk_B , pk_A , pk_{KDC},C-A_B, C_B

KDC: sk_{KDC} , pk_X, pk_{trust}

Après le protocole les deux parties connaissent les clefs publiques de leur partenaire. Si elles comptent reprendre une communication avec celui-ci, alors elles peuvent garder les clefs et elles n'ont plus besoin du distributeur de clefs.

c) Commentaires

Dans ce protocole, la séparation des rôles de certification des clefs et de distribution des clefs permet l'utilisation de plusieurs centres de certification

reconnus par le centre de distributions des clefs. Cela permettrait, par exemple, d'avoir un centre de distribution de clefs dans une faculté et un centre de certification dans chaque secrétariat de département.

La non-répudiation de l'origine est ici possible à condition que le centre de distribution des clefs réponde à certaines exigences. Le KDC doit entre autre, garder un historique des clefs publiques avec leur date de début et de fin de service. Nous nous référons à la fonction d'archivage qui est imputée à l'autorité de distribution des clefs. L'archivage doit comprendre les dates de validité des clefs ainsi que les révocations et les suspensions. Car une vérification en cas de litige est toujours possible. L'autorité de certification devra également répondre à d'autres exigences imposées par le législateur afin de permettre à celui-ci d'accepter les signatures électroniques.

La remarque à propos du dépôt de clefs est toujours valable.

2.6 Conclusion

Le long de ce chapitre, nous avons réussi à introduire dans des protocoles de communication les différentes notions que nous avions vues au chapitre précédent. Les trois rôles principaux des tiers de confiance ont été rencontrés ainsi que l'utilisation de la carte à puce et des dépôts de clefs. Mais il manque encore des éléments pour nous permettre la réalisation des tiers de confiance. Des éléments tels la gestion des clefs en mémoire secondaire ou la répudiation de clefs publiques sont encore floues et seront développées dans le chapitre suivant.

SPÉCIFICATION D'UNE INFRASTRUCTURE TTP

3. Spécification d'une infrastructure TTP

3.1 Introduction

Nous allons, dans ce chapitre, spécifier une infrastructure TTP basée sur le quatrième protocole vu au chapitre précédent. Cette infrastructure comprend plusieurs entités que nous retrouvons dans le schéma ci-dessous.

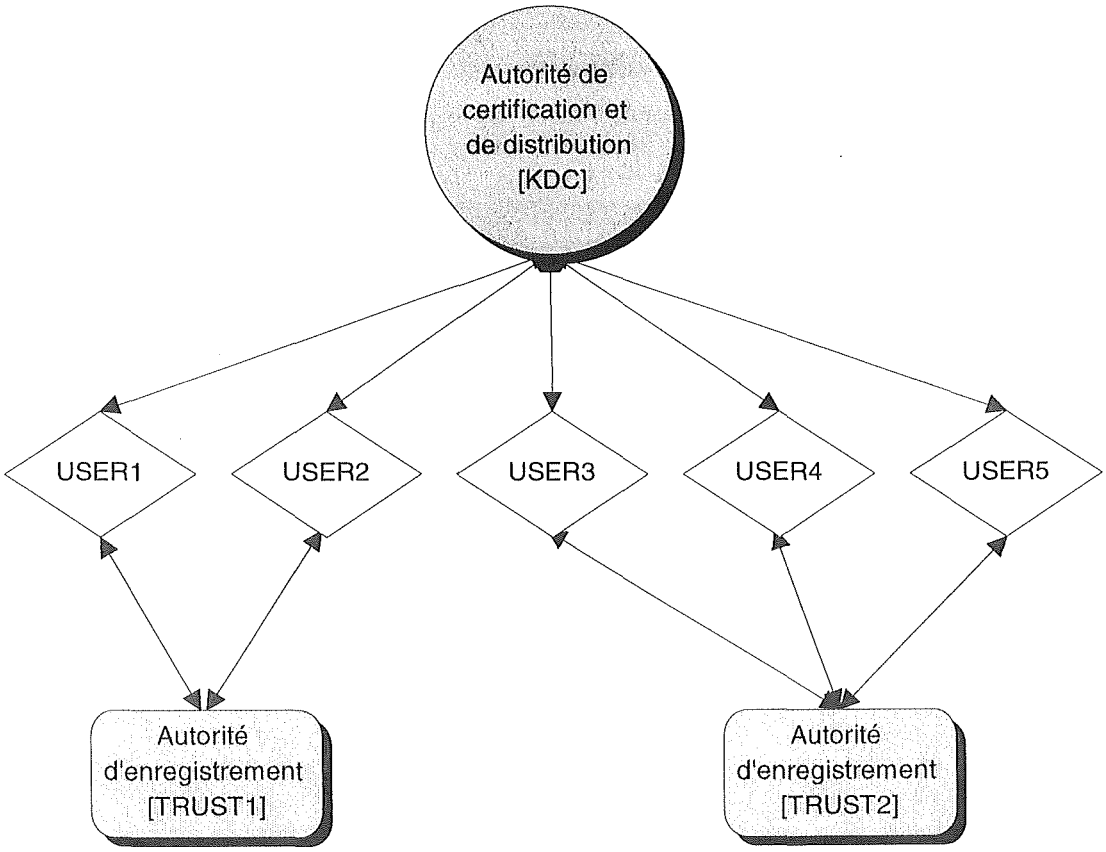

Figure 14 Le schéma d'une infrastructure TTP

Nous appellerons l'autorité de distribution des certificats et de certification KDC.

L'autorité d'enregistrement portera le nom de TRUST.

Nous supposons qu'il peut exister différents centres distributeur de clefs dans l'organisation, mais que tous sont de confiance (par exemple, dans une université, un par faculté). Nous supposons également qu'il peut y avoir plusieurs centres d'enregistrement « trust » dont le rôle est de certifier le lien entre la clef publique et l'identité.

Il est clair que l'architecture vise une petite organisation qui a entière confiance dans la partie certifiante et qui vise une très grande sécurité. Notons que nous avons décidé de ne stocker que les clefs publiques au lieu des certificats complets de type x.509. Ce choix se justifie par la taille de l'organisation qui permet d'utiliser des logiciels standards et par la facilité en cas d'implémentation.

Si nous voulions intégrer le système présenté dans un schéma plus global, voir mondial, il suffirait de stocker les certificats extérieurs au système; de faire certifier les clefs de l'autorité de certification par une autre autorité de certification supérieure (par exemple BelSign) et d'ouvrir des communications vers l'extérieur grâce au protocole x509. Cela demanderait peu de modifications.

Pour spécifier l'infrastructure TTP, nous devons donc spécifier trois logiciels qui sont les interfaces des trois entités. Nous allons donner une première explication en langage courant sur les structures de données persistantes et les fonctionnalités liées à ces interfaces. Ensuite, nous utiliserons le langage vu au cours de Méthodologie de Développement de Logiciels pour spécifier plus formellement ces logiciels. Les logiciels prévus ici permettent la non répudiation mais nous n'avons pas mis au point les fonctionnalités requises pour la mettre en œuvre.

3.2 Spécifications informelles

a) Structure de données persistantes pour la section utilisateur

Pour voir plus clair dans la structure de données, nous avons représenté

celle-ci par un schéma relationnel réalisé dans la figure ci-dessous.

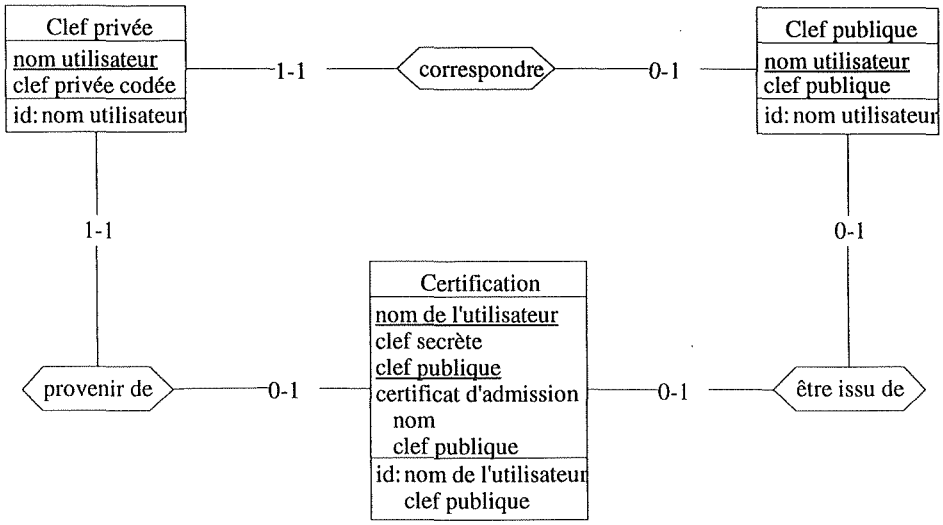

Figure 15. Structure de données chez un utilisateur

Mais la base de données est plus complexe, car elle n'est pas composée simplement de tables stockées sur le disque dure de la machine. La table contenant les clefs privées sera placée dans un cryptomodule pour améliorer la protection des clefs privées. La table est donc un ensemble de cartes à puce contenant chacune une clef privée chiffrée par un PIN (Personnal Identification Number) connu uniquement de l'utilisateur. Chaque carte à puce est personnelle. Pour chaque clef privée, il devra y avoir une et une seule clef publique correspondante. L'état clef publique est moins critique et pourra par conséquent se trouver sur le support physique de l'ordinateur. A l'initialisation du programme, l'état clef publique ne contiendra que la clef publique du ou des KDC et la clef publique correspondant à l'utilisateur. Le troisième état reprend les informations nécessaires à l'enregistrement et à la certification de la paire de clefs. Il est préférable que ces informations soient stockées sur un support amovible pour que l'utilisateur puisse se rendre facilement auprès de l'autorité d'enregistrement TRUST. Ce sera donc également un ensemble de cryptomodules. On peut prévoir de faire de ces deux derniers états un seul état mais nous avons préféré les garder séparés pour une plus grande clarté. Nous faisons une dernière remarque pour signaler que le choix de mettre le nom comme identifiant est dû au fait que nous pensions notamment utiliser la structure reprise dans PGP en cas

d'implémentation. (PGP étant un programme qui est utilisé pour gérer des chiffrements asymétriques et symétriques).

b) Structure de données persistante pour la section KDC

Nous avons également représenté les structures de données sur le schéma relationnel de la figure ci-dessous.

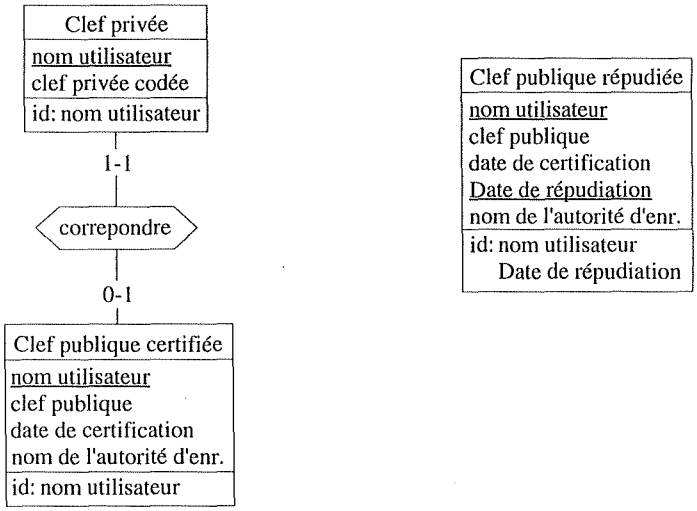

Figure 16 : Structure de données de KDC

L'état clef privée est également un ensemble de cartes à puce utilisé par le KDC. La table des clefs publiques certifiées reprendra toutes les clefs des adhérents au KDC ainsi que la date de certification et le nom de l'entité d'enregistrement qui atteste du lien clef-nom. Dans cette table, on retrouvera également les clefs du KDC et les clefs des autorités d'enregistrements (qui sont récoltées par un moyen sécurisé). Il est important d'archiver ces informations supplémentaires car, en cas de litiges, elles seraient primordiales. Pour cette même raison, ces informations se retrouvent dans la table des clefs répudiées ainsi que la date de répudiation. Une clef se trouvant dans la table des clefs publiques certifiées ne peut se trouver dans les tables de clef publiques répudiées et vice-versa.

c) Phase d'enregistrement et d'admission

La phase d'enregistrement n'est pas décrite à proprement parlé dans les spécifications formelles de la deuxième partie de ce chapître, car il s'agit plus d'une procédure administrative qu'informatique. L'utilisateur va générer sa paire de clefs qu'il va directement placer sur un support amovible. Ensuite, l'utilisateur va se rendre auprès d'un des centres d'enregistrement (TRUST) et va prouver son identité, par exemple à l'aide de sa carte d'étudiant ou de sa carte d'identité. Une fois son identité prouvée l'utilisateur donne sa clef publique à l'autorité d'enregistrement. Celle-ci va alors créer un message quelconque que l'utilisateur signera avec sa clef privée afin de prouver à l'autorité d'enregistrement qu'il possède bien la clef privée correspondant à la clef publique qu'il vient de donner. L'autorité d'enregistrement signe la clef et l'identité de l'utilisateur afin d'authentifier le lien entre eux. Il remet alors ce certificat d'admission qu'il vient de réaliser à l'utilisateur.

La procédure d'admission est très simple : une fois que l'utilisateur possède un certificat d'admission, il l'envoie à l'autorité de certification c-à-d KDC. Celui-ci vérifie la signature de TRUST et s'il n'a pas déjà une clef semblable dans ses tables. Si les vérifications sont positives, il certifie la clef en la plaçant dans sa table de clefs publiques certifiées avec la date et le nom de l'autorité d'enregistrement. La phase d'admission est maintenant effectuée. Voyons alors la communication en elle-même.

d) Phase de connexion

La phase de connexion commence par la demande de certificats du demandeur au KDC. Cette demande contient le nom de l'émetteur (Alice) et le nom du demandeur (BOB). Le KDC, après avoir reçu cette demande ,va rechercher certaines informations dans ses bases de données. Si les recherches sont fructueuses, il réalise deux certificats. En cas d'échec, il avorte la connexion et envoie un message d'erreur.

Ces certificats reprennent :

- L'identité du correspondant

- La clef publique du correspondant
- La clef de session chiffrée avec la clef publique du correspondant
- L'heure de création du certificat
- la signature de KDC

Une fois ces deux certificats réalisés, le KDC renvoie ceux-ci à Alice.

Une fois qu'Alice a reçu les deux certificats (le sien et celui de Bob), elle va vérifier la signature de KDC sur les deux certificats ainsi que les heures de création des certificats. Elle comparera alors le temps de réponse à un temps moyen avant d'accepter les certificats. Quand les Certificats sont acceptés, Alice déchiffre la clef de session avec sa clef privée qu'elle a recouvré sur sa carte à puce à l'aide de son PIN. Ensuite, elle stocke la clef publique de Bob dans sa base de données. Et elle réalise un message particulier qui contient la clef de session et le moment de création du message. Ce message est signé avec sa clef privée et chiffré avec la clef publique de Bob. Elle envoie alors les deux certificats et le message à Bob.

Bob pourra alors faire les vérifications de signature et les déchiffrements nécessaires (idem qu'Alice) afin d'obtenir la clef publique d'Alice qu'il va stocker et la clef de session. Il réalisera alors plusieurs vérifications comme les vérifications liées aux moments de créations des certificats et la comparaison entre la clef de session issue du message et celle issue du certificat. Il renverra alors les deux certificats ainsi qu'un message qui contient la clef de session et le time stamp. Avant de renvoyer ce message, il le signe et le chiffre avec la clef publique d'Alice.

Une fois qu'Alice reçoit les deux certificats et le message de Bob . Elle effectue quelques vérifications. Si ces vérifications sont positives, la session de communication peut commencer.

e) Session de communication

Alice va alors envoyer un message à Bob. Elle aura signé ce message et chiffré ce message signé avec la clef de session. Le fait d'utiliser une clef de session rend la communication plus sûre car si un message est décrypté les

messages de la session suivante restent sécurisés. Bob peut, après avoir déchiffré le message et vérifier la signature, répondre par le même procédé.

Pour clore la session, l'un des deux interlocuteurs va envoyer un message de clôture qui va lancer la destruction des clefs de sessions.

f) Répudiation des clefs

La répudiation des clefs peut être comprise à plusieurs niveaux. Il y a la répudiation de la clef d'un utilisateur, de la clef d'un TRUST et de la clef de KCD.

La répudiation de la clef de l'utilisateur peut avoir deux initiateurs : le KDC et l'utilisateur lui-même. L'utilisateur envoie un message signé qui demande la répudiation. Le KDC exécute alors la fonctionnalité de répudiation des clefs qui transfert la clef concernée de l'état clef certifiée à l'état clef répudiée.

La répudiation des clefs TRUST se réalise en appelant la fonctionnalité de répudiation de la clef d'utilisateur pour la clef TRUST. Ensuite, on applique la fonctionnalité de répudiation de la clef d'utilisateur à toutes les clefs d'utilisateurs qui ont été enregistrées par TRUST et certifiées après la divulgation de la clef. KDC enverra également un message a tous les utilisateurs.

La répudiation de la clef de KDC pose moins de problèmes, il lui suffit de changer de paire de clefs et d'avertir ses utilisateurs

g) Résolution de litige

Nous n'avons pas spécifié les fonctionnalités en cas de litige car il suffit de faire des accès aux différentes tables.

3.3 Spécifications formelles

a) Section utilisateur

Ce programme vise à envoyer et à gérer des documents chiffrés chez un

utilisateur. Ce logiciel sera donc l'interface entre l'utilisateur et le réseau. Il devra permettre l'adhésion de l'utilisateur dans le système et la communication sécurisée avec un tiers.

Etat clef privée

Interface :

Invariants globaux

si $\exists$ $sk_x \in$ Clefs-priv $\Rightarrow \exists$ $pk_x \in$ Clefs-publiq et pk_x est la clef publique correspondant à la clef privée pk_x

Structure de données:

Clefs-priv:= TBL[identif, (Ek_{ST}(SK))]

Identif:=STRING (un élément identifiant, en général adresse E-mail)

CLEF_DES=STRING (clef symétrique)

CLEF_PUB=LONG STRING (clef publique des correspondants connus)

CLEF_SEC= LONG STRING (clef secrète de l'utilisateur)

k_{ST}:= CLEF_DES

sk_X:= CLEF_SEC

pk_X:=CLEF_PUB

Propriétés

Initialisation

Card (Clefs-priv) $\neg$=0

Si on n'a pas de cryptomodule avec une identité et une clef privée authentifiée $\Rightarrow$ l'état est vide donc on ne sait rien faire.

Invariants locaux

not exist k_{st}, $k_{st'}$: $k_{st} = k_{st'}$

Les clefs pour garder les clefs privées doivent toutes être différentes.

Structure de données intermédiaires

k_{st}, $k_{st'}$: STRING

Etat clef publique

Interface :

Invariants globaux

Structure de données:

Clefs-publiq:= TBL[identif, (PK)]

Identif:=STRING (adresse E-mail)

pk := CLEF_PUB

Propriétés

Initialisation

Card(Clefs-publiq) =n

Si pk_X in Clefs-publiq et pk_X est la clef publique correspondant à l'identifiant X ⇒ X est un KDC

Quand on lance le programme, les seules clefs publiques qui se trouvent dans la base de données sont les clefs de l'utilisateur et de KDC. On doit toujours aller rechercher la clef publiques des autres correspondants pour une question de confiance dans les clefs. Les clefs du KDC ont été préalablement obtenues de façon sûre.

Invariants locaux

Structure de données intermédiaires

X= STRING

pk_X= CLEF-PUB

Etat certification

Interface :

Invariants globaux

Structure de données:

CERTIFICATION = SET[CP[pk, sk , ID, Certif-adm]]

Certif-adm = Ssk_{trust} (CP[ID, pk])

Ssk_{trust} = LONG STRING ⇒ LONG STRING cette fonction est simplement la fonction de signature avec la clef secrète. Nous plaçons en indice la clef utilisée.

ID :=STRING (adresse E-mail)

PK, = CLEF_PUB

sk_{trust} , sk = CLEF_SEC

N.B.: L'état certification comprend donc l'ensemble des cartes (ou des certificats) délivrées à l'utilisateur. Un certificat se trouve sur une carte à puce qui à été délivrée par une entité « trust ». Cette entité a vérifié que l'identité contenue dans le certificat est bien la même que celle de la personne à qui a été délivrée la carte et donc que le lien (clef - possesseur) est bien respecté. Sur la carte, on peut donc retrouver la clef publique, la clef secrète et le certificat d'admission.

Propriétés

Initialisation

not empty Certification

Quand on lance le programme pour la première fois, on doit se faire certifier.

Invariants locaux

$pk_{identif}$ est la clef publique correspondante à la clef secrète $sk_{identif}$.

Structure de données intermédiaires

$pk_{identif}$:= CLEF-PUB

$sk_{identif}$:= CLEF-SEC

Fonctionnalité de génération des certificats d'admission et des clefs

Interface :

Argument: Identif

Elém_aléatoire

TRUST

mess

Résultat: $CC_{Identif}$

Etat: Certification

Structure de données:

TRUST,Identif:=STRING

$CC_{Identif}$:= [CP[pk, pk, ID, Certif-adm]

Certif-adm = S_{SK} (CP[ID , pk])

ID:=STRING (adresse E-mail)

mess := LONG STRING

pk := CLEF_PUB

sk := CLEF_SEC

Elém_aléatoire: INTEGER

Propriétés

Règles de traitement

$pk_{identif}$, $sk_{identif}$:= GENERATION-PAIRE DE CLEF (Elém_aléatoire)

mess est reçu de Trust

Certif-$adm_{identif}$:= Certification (Identif, mess, $Ssk_{identif}$(mess),$pk_{Identif}$)

$CC_{Identif}$:= [CP[$pk_{identif}$, $sk_{identif}$, Identif, Certif-$adm_{identif}$]

La fonction de certification est au cœur du problème de l'authentification. En réalité, nous posons que l'utilisateur a enregistré sur un support les informations mises en argument. Il se rend avec ce support auprès d'une autorité de certification qui, après avoir vérifié l'identité de l'utilisateur et le fait qu'il a bien la clef secrète (par la vérification de la signature), va lui fournir un certificat d'admission. Les autorités de certification peuvent être, par exemple, pour une université, les secrétariats d'un département.

$CC_{Identif}$ In Certification

Structure de données intermédiaires

$pk_{identif}$:= CLEF-PUB

Certif-$adm_{identif}$:= Certif-adm

$sk_{identif}$:= CLEF-SEC

Fonctionnalité demande de connexion au TTP

Interface :

Argument: KDC

CC_A

TRUST

Elém_aléatoire

Résultat: OK_adhésion

Etat: Certification

Clefs-priv

Clef-publiq

Structure de données:

KDC, TRUST:=STRING

OK_adhésion := BOOLEAN

Certification :=SET[CP[pk, sk, ID, Certif-adm]]

CC_A:= [CP[pk, sk, ID, Certif-adm]

Certif-adm := Ssk_{TRUST} (CP[ID, pk])

ID := STRING (adresse E-mail)

pk := CLEF_PUB

sk_{trust}, sk := CLEF_SEC

Elém_aléatoire :=INTEGER

Propriétés

Règles de traitement

SEND-MAIL Ssk_{TRUST} (A, pk_A),TRUST TO KDC

RECEIVE-MAIL $C1_{A'}$ FROM KDC

CC_A in Certification

Certif-adm(CC_A) = Ssk_{TRUST}(A, pk_A)

pk_{KDC} = clefs-publiq[KDC]

OK_adhésion = true si {Vpk_{KDC} ($C1_A$))=A, pk_A, T_{0A} and

T_{0A}<T+σ }

T_{0A} est un time stamp de création du certificat. Nous vérifierons une condition du type T_{0A}<T+σ où T est le moment de l'envoi de l'E-mail et où σ est un temps maximal de réponse.

NB C_A est obtenu de KDC par un échange de E-Mail

sk_A :=sk(CC_A)

pk_A :=sk(CC_A)

A :=ID(CC_A)

$k_{ST\ A}$:= Génération-CLEF-DES (Elém_aléatoire)

si OK_adhésion alors {Clefs-priv = Append(Clefs-priv, $Ek_{ST\ A}$ (A,sk_A))and

Clefs-publiq = Append(Clefs-publiq,(A,pk_A))}

MESSAGE D'ERREUR

Structure de données intermédiaires

T_{0A},T,σ:=TIME

A:=STRING

pk_{KDC} , pk_A := CLEF_PUB

sk_A :=CLEF_SEC

$C1_A$:=S_{SKkdc}(ID, pk, T)

Fonctionnalité demande de connexion avec interlocuteur

Interface :

Argument: KDC

A

B

Résultat C_A,C_B

Etat Clefs-publiq

Clefs-priv

Structure de données:

A, B, KDC :=STRING

Ident =STRING

C_A,C_B:=Ssk_{KDC}(KDC, Ident, Epk_{ident}(tk), pk_{ident}, T),

T :=TIME (moment de la création du certificat)

pk_{ident}:=CLEF_PUB

tk:= CLEF_SESS(clef de session symétrique)

CLEF_SESS=LONG STRING

Propriétés

Règles de traitement

SEND-MAIL A, B,TO KDC

RECEIVE-MAIL C_A ,C_B FROM KDC

C_A,C_B Sont reçus de KDC qui les génère et les envois par E-Mail d'après

les informations reçues

pk_{KDC} = Clefs-publiq [KDC]

Si { Vpk_{KDC} (C_B) = B, Epk_B (tk), pk_B, T1 (on ne peut pas vérifier Epk_B(tk) ni pk_B)

and Vpk_{KDC} (C_A) = A, Ekp_A (tk), pk_A, T2

and T1=T2

and pk_A = Clefs-publiq[A]

and T1 <T+σ } alors Clefs-publiq = Append(Clefs-publiq,(B, pk_B))

sinon MESSAGE d'ERREUR

T1 et T2 sont des timbres temporels (time stamp) de création du certificat nous vérifierons une condition du type T1<T+σ où T est le moment de l'envoi de l'E-mail et où σ est un temps maximal de réponse.

Structure de données intermédiaires

pk_{KDC} , pk_B, pk_A =CLEF_PUB

T1, T2, T, σ:= TIME

Fonctionnalité ouverture de session

Interface :

Argument : C_A

C_B

X_{AB}

B

$K_{ST\ A}$

Résultat : X_{BA}

OK-Connexion

Etat : Clefs-priv

Clefs-publiq

Structure de données:

B:=STRING

Ident = STRING

C_A, C_B := Ssk_{KDC} (KDC, Ident, Epk_{ident} (tk), pk_{ident}, T),

X_{AB}, X_{BA} := $Epk_{ident}(Ssk_{ident}(tk, T_{ident}))$

T , T_{ident}:=TIME

pk_{ident}:=CLEF_PUB

sk_{ident}:=CLEF_SEC

tk:= CLEF_SESS(clef de session symétrique)

OK-Connexion := Boolean

$k_{ST\,A}$:= CLEF-DES

Propriétés

Règles de traitement

SEND-MAIL C_A ,C_B ,X_{AB}TO B

RECEIVE-MAIL C_A ,C_B ,X_{BA} FROM B

$Vpk_{KDC}(C_A)$=KDC , A, $Epk_A(tk_{AB})$, pk_A; T

$tk_{AB} = Dsk_K(Epk_A(tk_{AB}))$

$Vpk_{KDC}(C_B)$=KDC, B ,$Epk_B(tk_{AB})$,pk_B ; T

$X_{AB} = Epk_B(Ssk_A(tk_{AB}, T_A))$

Si {$Vpk_B(Dsk_a(X_{BA}))=tk_{AB}$, T_B and

$T_B > T_A + \sigma$ } alors OK-Connexion :=true

sinon MESSAGE D'ERREUR

T_A est le moment de l'envoi de l'E-mail et σ est un temps maximal de réponse.

Pk_{KDC} = Clefs-publiq[KDC]

sk_A := $Dk_{ST\,A}$(Clefs-priv[A])

Structure de données intermédiaires

T_B,T_A,σ :=TIME

sk_A :=CLEF_SEC

tk_{AB} := CLEF_DES

pk_{KDC} := CLEF_PUB

Fonctionnalité réponse à une ouverture de session

Interface :

Argument : C_A

C_B

X_{AB}

A

$k_{ST\,B}$

Résultat : X_{BA}

OK-Connexion

Etat : Clefs-priv

Clefs-publiq

Structure de données:

A, B ,ident :=STRING

C_A,C_B := Ssk_{KDC}(KDC, ident, Epk_{ident}(tk), pk_{ident}, T'),

X_{AB}, X_{BA} := Epk_{ident}(Ssk_{ident} (tk, T_{ident}))

T, T_{ident}:=TIME

pk_{ident}:=CLEF_PUB

sk_{ident}:=CLEF_SEC

tk:= CLEF_SESS(clef de session symétrique)

Propriétés

Règles de traitement

RECEIVE-MAIL C_A ,C_B,X_{AB} FROM B

pk_{KDC} =Clefs-publiq[KDC]

$Vpk_{KDC}(C_B)$= KDC, B, Epk_B (tk_{AB}) ,pk_B ,T'

tk_{AB}= $Dsk_B(Epk_b(tk_{AB}))$

X_{BA}= $Epk_A(Ssk_b($ tk_{AB},T_A))

T_B= temps au moment de la réalisation

Si { $Vpk_{KDC}(C_A)$=KDC, A, $Epk_A(tk_{AB})$,pk_A;T'

Vpk_a (Dsk_B(X_{AB}))=tk_{AB}' , T_A and

TK_{AB}'=TK_{AB} and

T_B>T_A+σ } alors OK-connexion :=true

sinon MESSAGE D'ERREUR

T_A est le moment de l'envoi de l'E-mail et σ est un temps maximal de réponse.

si OK-Connexion ⇒SEND-MAIL C_A ,C_B ,X_{BA} TO A

Structure de données intermédiaires

$T_{B'}$,$T_{A'}$,σ :=TIME

KDC, B, :=STRING

tk_{AB} , $tk_{AB'}$:= CLEF_DES

pk_B ,pk_A ,pk_{KDC}

Fonctionnalité discussion-session

Interface :

Argument A

B

TK

Message

Résultat Réponse

Etat Clefs-priv

Clefs-publiq

Structure de données:

B ,A:= STRING

TK:= CLEF-SESS

Message , Réponse := FILE

Propriétés

Règles de traitement

sk_A = $Dk_{ST\ A}$(Clefs-priv[A])

pk_B := Clefs-publiq [B]

Message-chiffré=E_{TK}(Ssk_A(Message))

SEND-MAIL Message-chiffré TO B

RECEIVE-MAIL Réponse-chiffrée FROM B

Réponse =Vpk_B (D_{TK}(Réponse-chiffrée))

Structure de données intermédiaires

Message-chiffré, Réponse-chiffrée :=FILE

sk_A := CLEF_SEC

$k_{ST\ A}$:= CLEF_DES

N.B. Nous n'avons pas spécifié la fonction de réponse qui est strictement l'inverse de cette dernière fonction.

Fonctionnalité fin de session du demandeur

Interface :

Argument: A

B

TK

Résultat: « destruction de TK »

Structure de données:

B,A:=STRING

TK:= CLEF-SESS

Propriétés

Règles de traitement

SEND-MAIL E_{TK}(A,B, »stop ») TO B

destruction de TK

Structure de données intermédiaires

Fonctionnalité fin de session du récepteur

Interface :

Argument: TK

Résultat: « destruction de TK »

Structure de données:

TK:= CLEF-SESS

Propriétés

Règles de traitement

RECEIVE-MAIL E_{TK}(B,A, »stop ») FROM B

destruction de TK

Structure de données intermédiaires

B,A:=STRING

Fonctionnalité de répudiation de clefs

Interface :

Argument KDC

A

pk_A

Structure de données:

KDC,A:= STRING

pk_A :=CLEF-PRIV

Propriétés

Règles de traitement

SEND-MAIL Ssk_A(A, pk_A, »répudiation »)·TO KDC

Clefs-priv = Remove(Clefs-priv,(A, $Ek_{st\ A}$ (sk_A))

Structure de données intermédiaires

$K_{st\ A}$= CLEF-DES

b) Section Trusted Third Party

Cette section décrit les fonctionnalités de l'interface du KDC. Rappelons que le KDC est une autorité de certification et de distribution des clefs.

Etat clef privée

voir Section Utilisateur

Etat clef publique certifiée

Interface :

Invariants globaux

Structure de données:

Clefs-publiq := TBL[identif, CP($pk_{identif}$, Date-in)]

Identif :=STRING (adresse E-mail des utilisateurs)

pk_{ident} := CLEF_PUB(clef publique des utilisateurs)

Propriétés

Initialisation

Card(Clefs-publiq) =1+m

si pk_X in Clefs-publiq et pk_X correspond à l'utilisateur X $\Rightarrow$ X est le KDC ou X est un des m TRUST (autorité de certification)

Les clefs des autorités de certification doivent être obtenues de façon très sécurisée de préférence sans passer par le réseau ouvert.

Invariants locaux

Structure de données intermédiaires

Etat clef publique répudiée

Interface :

Invariants globaux

Si pk_A in $pk_{identif}$ (Clefs-repud[A]) alors pk_A not = $pk_{identif}$ (Clefs-publiq [A])

Structure de données:

Clefs-repud := TBL[identif, SET (CP($pk_{identif}$,Date-in, Date-out))]

Identif :=STRING (adresse E-mail des utilisateurs)

$pk_{identif}$:= CLEF_PUB(clef publique des utilisateurs)

Propriétés

Initialisation

Card(Clefs-repud) = 0

Invariants locaux

Structure de données intermédiaires

Fonctionnalité mise en certification d'une clef

Interface :

Argument: A

Ssk_{TRUST} (A, pk_A)

TRUST

Résultat: OK_adhésion_ttp

C_A

Etat: Clefs-priv

Clefs-publiq

Structure de données:

TRUST:=STRING

OK_adhésion_ttp :=BOOLEAN

A:=STRING (adresse E-mail)

$pk_{A,}$:= CLEF_PUB

C_A=Ssk_{KDC} (KDC ,A , pk_A,T_{A0})

Propriétés

Règles de traitement

RECEIVE -MAIL Ssk_{TRUST}(A, pk_A), TRUST FROM A

SEND -MAIL C_A TO A

sk_{KDC} =Clefs-priv[KDC]

pk_{TRUST}=$pk_{identif}$(Clefs-publiq[TRUST])

si Vpk_{TRUST} (Ssk_{TRUST}(A,pk_A))=A,pk_A⇒ OK_adhésion =true

T_{0A} est un time stamp de création du certificat. Nous vérifierons une

condition du type :

si OK_adhésion $\Rightarrow$ Alors $C_A = Ssk_{KDC}(KDC, A, pk_A, T_{0A})$

{Clefs-publiq = Append(Clefs-publiq,(A, (pk_A,T_{0A})))}

Sinon message d'erreur

Structure de données intermédiaires

T_{0A},T,σ:=TIME

KDC,A:=STRING

Pk_{KDC} := CLEF_PUB

Pk_{TRUST} := CLEF_PUB

Fonctionnalité mise en connexion de deux interlocuteurs

Interface :

Argument: Elém_aléatoire

A

B

Résultat: C_A,C_B

Etat: Clefs-publiq

Clefs-priv

Structure de données:

A, B, KDC, ident :=STRING

C_A,C_B:=Ssk_{KDC} (KDC,ident, Epk_{ident}(TK), pk_{ident}, T'),

T':=TIME

pk_{ident} :=CLEF_PUB

TK:= CLEF_SESS(clef de session symétrique)

sk_{KDC} = CLEF_SEC

Elém_aléatoire := INTEGER

Propriétés

Règles de traitement

RECEIVE-MAIL A, B FROM A

si A in Clefs-publiq and B in Clefs-publiq et qu'il n'y a pas eu de message

d'erreur SEND-MAIL C_A ,C_B TO A

sk_{KDC}= Clefs-priv[KDC]

T' = temps au moment de la création du Certificat

pk_B= $pk_{identif}$ (Clefs-publiq[B])

si B not in Clefs-publiq⇒{ MESSAGE-ERREUR and SEND-MAIL MESSAGE-ERREUR TO A}

pk_A= $pk_{identif}$ (Clefs-publiq[A])

si A not in Clefs-publiq⇒{ MESSAGE-ERREUR and SEND-MAIL MESSAGE-ERREUR TO A}

tk_{AB} =Génération de clef de session (Elém_aléatoire)

C_B =Ssk_{KDC}(KDC, B, $E_{PKb'}(tk_{AB})$, pk_B, T')

C_A =Ssk_{KDC}(KDC, A, $E_{PKb'}(tk_{AB})$, pk_A, T')

Structure de données intermédiaires

pk_{KDC}, pk_B , pk_A =CLEF_PUB

tk_{AB} := CLEF-SES

Fonctionnalité de répudiation de clef d'interlocuteurs

Interface :

Argument Ssk_A (A,pk_A, »répudiation »)

Etat: Clefs-publiq

Structure de données:

A:= STRING

PK_A:=CLEF-PUB

SK_A:=CLEF-SEC

Propriétés

Règles de traitement

RECEIVE-MAIL Ssk_A (A, pk_A, »répudiation »)·FROM A

pk_A := $pk_{identif}$ (Clefs-publiq[A])

si Vpk_A (Ssk_A(A, pk_A, »répudiation »))= A,PK_A, »répudiation »

⇒ Clefs-publiq = Remove(Clefs-publiq,(A,(PK_A, T_{0A})))

Clefs-repud = insert (Clefs-repud,(A,(PK_A, T_{0A}, T_{1A})))

T_{1A} temps au moment de la répudiation. Ce temps est imposé par le KDC pour éviter les fraudes.

Fonctionnalité de répudiation de sa clef

Interface :

Argument: pk_{KDC}

Etat: Clefs-publiq

Clefs-priv

Structure de donnée:

pk_{KDC} = CLEF-PUB

Propriétés

Règles de traitement

Clefs-priv = Remove(Clefs-priv,(KDC, E_{Kst} (SK_{KDC}))

K'st =génération de clef des

sk_{KDC} = génération de clefs secrètes RSA

pk_{KDC} = génération de clefs privées RSA

Clefs-priv = Append (Clefs-priv,(KDC, $E_{K'st}$ (sk'_{KDC}))

Clefs-publiq = Append (Clefs-publiq,(KDC, (pk'_{KDC} ,T_{1KDC}))

Clefs-repud = insert (Clefs-repud,(KDC,(PK_{KDC}, T_{0KDC}, T_{1KDC})))

$\forall$ A in Clefs-publiq |SEND-MAIL KDC, PK'_{KDC} TO A

Publication de la clef par un élément extérieur (ex. un journal)

T_{1KDC} Moment de l'enregistrement des clefs.

3.4 Conclusion

Dans ce chapitre, nous avons décrit une infrastructure de tiers de confiance pour une petite organisation. Cette infrastructure permet l'authentification, la non-répudiation et la confirmation. Pour une petite organisation ces fonctions semblent remplir la plupart des besoins.

ASPECT JURIDIQUE DE LA SÉCURITÉ DE L'INFORMATION

4. Aspects juridiques de la sécurité de l'information

4.1 La légalité de la confidentialité

a) Introduction

Le point de vue qui va être développé ici, est centré sur l'usage des techniques cryptographiques pour assurer la confidentialité des télécommunications. La plupart des informations sont tirées d'un rapport juridique réalisé par le groupe Belinfosec.

La discussion actuelle tourne surtout autour de l'opposition entre l'utilisation de la cryptographie et la loi sur les écoutes des communications et des télécommunications privées. Selon la loi, il faudrait trouver un moyen de laisser la possibilité aux agents légaux d'écouter une communication chiffrée sans rendre cette communication facilement décryptable par un quidam quelconque.

Il est clair que, vu la complexité des méthodes de chiffrement et de l'importance des protocoles, un minimum de techniques doivent être maîtrisées afin de pouvoir permettre de légiférer sur ce sujet. Nous ne reviendrons cependant pas sur ces techniques qui ont été vues dans les chapîtres précédents.

Un premier point à étudier est de voir quel est l'état actuel de la loi dans ce domaine. Ensuite, nous essayerons de voir quelques voies de solutions techniques.

b) Etat de la législation actuelle en matière de chiffrement.

Le chiffrement est une manière efficace d'assurer la sécurité des documents numériques aussi bien dans leur stockage que dans leur transfert. Mais, ces moyens de sécurisation sont-ils adaptés dans le cas d'écoutes

pratiquées dans le cadre de la loi du 30 juin 1994 ?

Il est intéressant de voir et de commenter cette loi ainsi que d'autres articles se rapportant au même sujet[2] :

Loi du 30 juin 1994 relative à la protection de la vie privée contre les écoutes, la prise de connaissance et l'enregistrement de communications et télécommunications privées.

Cette loi s'inscrit dans le prolongement de la loi du 8 décembre 1992 relative à la protection de la vie privée à l'égard des traitements de données à caractère personnel et a pour objectif de consacrer le principe de la protection de la vie privée contre les écoutes tout en autorisant ces dernières à titre exceptionnel et ce aux fins de répondre à la nécessité de la lutte contre le terrorisme et la grande criminalité.

La faculté d'effectuer des écoutes doit être considérée comme une dérogation au principe général d'interdiction de pratiquer des écoutes. Les cas les justifiants ainsi que la procédure dans lesquelles elles doivent être réalisées sont strictement définis par la loi.

Notons toute fois que l'écoute, la prise de connaissance ainsi que l'enregistrement de communication et de télécommunication doit avoir lieu pendant leur transmission et que toutes les communications ou télécommunications faisant l'objet d'une mesure de surveillance doivent être enregistrées, transcrites et, le cas échéant, traduites.

Le fait que l'enregistrement ne soit pas directement compréhensible est envisagé ici dans le cas de langues étrangères. A priori, il n'y a pas de raison de considérer une conversation chiffrée de manière différente. On considérerait, dès lors, le déchiffrement (avec les clefs) comme une traduction. Mais pour cela il est nécessaire de fournir les clefs de chiffrement. Par quel moyen l'état doit-il obtenir ces clefs ? Doit-il les exiger a priori en les rassemblant à l'avance au cas où il en aurait besoin ou les exiger a posteriori? La réglementation des télécommunications ne répond toujours pas à cette question.

Afin que la loi sur les écoutes ne soit pas vidée de son sens, il était indispensable que parallèlement, des adaptations au niveau de la réglementation des télécommunications aient lieu. Tel est l'objet de la loi du 21 décembre 1994.

Loi du 21 mars 1991 portant réforme de certaines entreprises publiques économiques.

2.1 Article 70 bis

Le Roi se voit confier la mission de fixer, par arrêté délibéré en Conseil des

[2] Les textes de loi sont en italique

Ministres, les moyens par lesquels Belgacom et les exploitants des services non réservés, qu'il désigne, doivent permettre, le cas échéant et éventuellement conjointement, l'application de la loi du 30 juin 1994 sur les écoutes.

Cette adaptation s'inscrit dans le prolongement de l'article 90 quater, § 2 du code d'instruction criminelle (introduit par la loi sur les écoutes), qui prévoit la possibilité, pour le juge d'instruction, de requérir le concours technique des opérateurs de réseaux pour l'exécution d'une mesure d'écoute.

Les opérateurs ont donc l'obligation de faire en sorte que, du point de vue technique, leurs infrastructures, appareils et services puissent être écoutés.

Dans cette réglementation, on ne fixe rien à propos des caractéristiques des «appareils », de plus les coûts de surplus de frais occasionnés reviennent à l'opérateur, donc à ses clients. L'étendue des responsabilités n'est pas bien précisée dans la loi. Nous restons dans le flou quant aux possibilités d'un particulier de chiffrer son courrier électronique. Si son provider a la capacité de décrypter ses messages, il a le droit d'utiliser le chiffrement mais peut-être pas l'intérêt. Nous espérons trouver des informations supplémentaires sur les moyens techniques dans l'article suivant.

Article 95 alinéas 1er

L'article 95 alinéa 1er, figurant au chapître VIII, «appareils terminaux »prévoit une nouvelle possibilité pour le Ministre, sur proposition de l'institut belge des services postaux et des télécommunications, de retirer un agrément ou imposer une interdiction de maintenir un raccordement à l'infrastructure publique de télécommunications lorsqu'il s'avère que l'appareil rend inefficace les moyens permettant l'application de la loi du 30 juin 1994 sur les écoutes

Cet article cite les appareils terminaux mais sans en donner une définition précise. Un équipement muni d'aucun hardware de cryptographie mais bien de software peut-il être touché par ces mesures ? Il semblerait que non. De plus, l'appareil de cryptographie peut ne pas être relié au réseau. La définition européenne (à laquelle la loi belge s'est adaptée) prévoit que le système de connexion consiste en fils métalliques, liaisons radioélectriques, systèmes optiques ou tout autre système électromagnétique.

L'agrément d'un appareil est quelque chose de précis dans la loi. Au travers de cet agrément le législateur aurait pu faire apparaître les contraintes techniques nécessaires afin de rendre efficace la loi sur les écoutes

téléphoniques. Mais, malgré cela, le législateur a introduit une disposition arbitraire puisqu'aucune norme technique justifiant un retrait n'a été spécifiée. La loi dans son état actuel peut être comprise comme étant réalisée dans le but de restreindre l'utilisation de moyens visant la confidentialité des communications.

Le terme «rend *inefficaces* » est mal choisi car il est très vague. Le cas où les communications ne sont pas directement compréhensibles est prévu dans la loi puisque le législateur parle de traduction. Le législateur ne parle pas du cas où la traduction est impossible ou prendrait 10000 ans. De plus, la compréhension est toujours floue. Prenons l'exemple suivant : Si nous envoyons un fichier compressé à quelqu'un. Est-ce considéré comme un moyen de rendre inefficace une procédure d'écoute? Actuellement, nous ne pouvons pas répondre à cette question de façon précise.

Voyons maintenant le point de vue européen. La convention européenne des droits de l'homme (C.E.D.H) prévoie que les restrictions à la liberté de correspondance doivent reposer sur un système de limitation matérielle. Pour qu'une restriction soit conforme aux dispositions de la convention, elle doit (1) être prévue par la loi, (2) constituer une mesure nécessaire dans une société démocratique et (3) être conçue dans l'intérêt d'un «objet de droit »protégé. L'ingérence est donc autorisée pour autant qu'elle soit nécessaire et qu'il y ait une menace contre la sécurité nationale ou la prévention des infractions pénales. Cette limitation est devenue un véritable critère de limitation des écoutes.

c) Les voies de solutions

Il n'y a pas de solutions simples à ce problème mais il existe cependant des voies vers des solutions acceptables. Une première voie dont nous avons déjà parlé est le choix d'un système de dépôt de clefs (ou key escrow) généralisé. C'est cette voie (appelée également contrôle a priori)que semble vouloir prendre le gouvernement américain qui a par le passé essayé d'imposer un système de dépôt de clefs hardware appelé Clipper. La question essentielle du point de vue juridique se rapportant au dépôt de clefs

est de savoir si le législateur ou le Roi pourrait imposer la remise des clefs. Il semble qu'au regard de la loi belge qui ne permet pas les écoutes préventives, il sera difficile d'imposer la remise des clefs.

L'autre voie vers laquelle semble se tourner le groupe Belinfosec est le déchiffrement a posteriori qui permet au gouvernement d'exiger les clefs de déchiffrement à l'aide d'un mandat. Mais les autorités judiciaires auront préalablement enregistré les communications. Dans le cas d'un refus de remise des clefs, le suspect pourra se voir attaquer en justice.

Une dernière voie est citée dans le document de Belinfosec. Cette solution vise à limiter le niveau de chiffrement pour rendre tout document décryptable par l'état. Mais s'il est décryptable par l'état, ne l'est-il pas par tout le monde?

4.2 La légalité de la signature électronique

[Davi97]

a) Introduction

Il n'est pas facile de donner des textes de loi essentiellement basés sur la signature. Car la signature est le vestige d'un véritable système de signes d'identité dont elle se détache au XVIIeme siècle. Il est intéressant de remarquer la place de la signature manuscrite qui est un moyen d'identifier, où la singularité de l'être est inscrite dans le signe. Tandis que les autres signes d'identité médiévaux, comme les sceaux, privilégient la détermination de l'individu par le réseau de ses appartenances. L'identification dans les réseaux électroniques se rapproche du système médiéval d'identification en ce qu'il se réfère à des caractéristiques extérieures à l'individu dont attestent des tiers. Nous remarquons que la ressemblance ne s'arrête pas là car nous retrouvons également le réseau d'appartenance dans la hiérarchie des certifications. Nous devons donc considérer la signature électronique de manière différente par rapport à la signature manuscrite, car il y a perte du lien physique au signataire. C'est ce lien qui avait permis à la signature manuscrite de supplanter les autres signes d'identité. Un autre avantage de

la signature électronique est qu'elle inclut l'ensemble du document qu'elle signe (authenticité d'un document).

b) Les fonctions de la signature

La signature a deux fonctions d'un point de vue légal. La première de ces fonctions est de permettre l'identification de l'auteur de l'acte. Elle confère à l'acte son authenticité. La deuxième fonction de la signature est d'exprimer la volonté du signataire. La signature apparaît comme l'extériorisation d'une volonté interne.

Voyons maintenant la définition légale de la signature.

1°La signature doit être manuscrite ' Elle implique un mouvement corporel (de la main en principe), et elle ne peut être remplacée par la reproduction au moyen d'un timbre humide, par une griffe ou par un cachet '.

2°La signature doit consister dans l'apposition autographe de son nom. En Belgique, *La signature est la marque manuscrite par laquelle le testateur revèle habituellement sa personnalité au tiers.*

c) La signature vue comme une numérotation

On peut considérer que la signature électronique revient à numéroter les individus. Ce qui pose un problème d'acceptation sociale [VERT94]de la signature électronique. L'efficacité de l'identification par le numéro est très grande. Elle présente en effet une possibilité de fonctionnement total : elle saisit l'irréductibilité individuelle, de même qu'elle intègre l'expansion croissante du groupe. Elle combine les deux grandeurs de l'identité : l'attachement au groupe et la singularité au sein du groupe. L'absence de reconnaissance dans un numéro pour un individu constitue le point névralgique des réseaux télématiques. On n'est jamais sûr que la personne qui utilise le terminal (au moyen d'une carte ou d'un PIN) est bien l'utilisateur légitime. Combien de personnes n'ont déjà prêté leur mot de passe pour ce logger ?

d) L'identification dans un système fermé

La technique des systèmes d'information fermés offre depuis longtemps des techniques d'identifications utilisées pour des réseaux fermés. Ce sont ces techniques qui sont utilisées pour des systèmes de distributeur de billets. Dans de tels systèmes, nous remarquons que la signature électronique est une nécessité impérieuse pour garantir le paiement. Mais le lien entre l'individu et sa signature peut facilement être garanti par la fermeture de réseaux (exemple de l'Echange de Données Informatisées sur des Réseaux à Valeur Ajouté[PARI97]). La reconnaissance de la valeur juridique de ces techniques d'authentification peut prendre trois orientations. Une première orientation se fonde sur l'analyse fonctionnelle de la notion de signature. Certains prétendent que l'on peut, dès à présent, reconnaître aux signatures digitales les mêmes valeurs que l'on reconnaît à la signature manuscrite, pour autant que les fonctionnalités d'identification de l'auteur et d'adhésion au contenu soient rencontrées. Ce qui serait, semble-t-il, le cas grâce à la signature à l'aide d'une clef privée (adhésion au contenu) et au certificat qui prouve le lien entre la clef publique correspondante et l'identité du signataire (identification du signataire). La deuxième approche est contractuelle. Les contractants s'entendent sur la valeur qu'ils attribuent aux signatures digitales. La troisième orientation possible est la voie législative. Le législateur pourrait redéfinir plus largement le concept de signature de façon à pouvoir y inclure des formes de signatures électroniques. Prenons comme exemple l'article 2827 du code civil du Québec qui donne une définition plus large de la signature permettant l'élargissement de la définition aux signatures électroniques.

Article 2827 du code civil du Québec : la signature consiste dans l'apposition qu'une personne fait sur un acte de son nom ou d'une marque qui lui est personnelle et qu'elle utilise de façon courante, pour manifester son consentement. Le législateur pourrait également définir une signature électronique et fixer la valeur de cette signature. Le problème fondamental à l'élargissement de ces possibilités aux réseaux ouverts est l'absence de véritable code

d'identification.

e) La signature dans un système ouvert

Un système ouvert est défini comme un système dans lequel aucune entité administrative ou légale ne contrôle les activités de communication, le stockage d'information ou les utilisateurs. L'absence de contrôle constitue une donnée majeure comme le fait qu'il peut exister plusieurs intervenants étrangers. Il est clair qu'il ne peut être question de transactions commerciales dans un réseau ouvert qui ne se serait pas donné les moyens de sécurité pour gérer les risques inhérents à son ouverture.

D'un point de vue juridique la sécurité d'un réseau ouvert se réfère à :

- L'authentification d'un utilisateur pour éviter l'usurpation d'identité et la répudiation d'une volonté exprimée.
- L'intégrité du message.
- La préservation des traces de la transaction pour valoir preuve.

Dans un système ouvert, la sécurité est contenue à même le message par les moyens de signature électronique. La signature électronique est appelée à jouer seule la fonction d'identification d'un partenaire, laquelle doit avoir lieu en amont du contrat. La signature électronique apparaît comme un processus incontournable afin de permettre l'expression du consentement d'une personne particulière. Elle peut donc participer à la formation d'un contrat. Il est en tout cas du ressort des parties de ramener l'exigence de signature dans le domaine de la formation de contrat et de convenir que l'échange contraignant ne peut résulter que de l'expression de la volonté signée électroniquement. Un problème majeur reste en vigueur: l'identification des acteurs dans un environnement ouvert. L'utilisation du chiffrement à clef publique est la solution qui ressort actuellement. Mais des problèmes de responsabilité se posent. Par exemple, la responsabilité de la conservation des clefs secrètes ou de l'identification n'est pas fixée.

On est obligé de créer un lien sûr entre une clef publique et l'identité de son possesseur. L'utilisation du chiffrement asymétrique suppose le recours à des mécanismes de contrôle visant à s'assurer que la clef rendue publique

est bien celle de la personne qui s'en prétend titulaire. Un tel contrôle est possible grâce à des entités qui produisent des certificats attestant de l'existence d'un lien entre une clef publique et tel utilisateur déterminé.

Les autorités de certification sont des tiers de confiance qui sont chargés par un ou plusieurs utilisateurs de créer et d'attribuer des clefs publiques et leurs certificats. Leur mission première est la certification des clefs publiques. Suite à cette vérification, l'autorité de certification émet un certificat de clef publique contenant notamment le degré d'exactitude des affirmations contenues dans le certificat. La deuxième mission d'une autorité de certification est de veiller à la constitution et à l'actualisation du répertoire contenant les certificats de clefs publiques qu'elle a émis. Elle aura en charge la gestion de la suspension, de la révocation et de l'expiration des certificats.

Selon le projet de loi allemand, la durée de sécurité d'une signature électronique est limitée à 5 ans et pour conserver sa valeur probante le document devra être résigné avant l'expiration de cette période de 5 ans. Le premier champ de responsabilité de l'autorité de certification réside dans l'émission du certificat. Car dans ce cas elle affirme que les informations contenues dans le certificat sont exactes et complètes. La responsabilité d'une autorité de certification peut également être engagée si celle-ci ne procède pas à la suspension ou à la révocation.

Pour mettre au point la législation de la signature électronique, il convient de définir les mécanismes de signature électronique dans des textes de lois. Mais de tels textes devraient disposer de la souplesse nécessaire pour répondre aux changements rapides et incessants du commerce électronique. Il serait également intéressant de prendre des décisions quant aux standards à utiliser. Il faudrait donc adapter le système juridique en vue de la réception de mécanismes de signatures électroniques. On peut plaider pour une disposition nouvelle assimilant à une signature manuscrite des signatures électroniques répondant à certaines normes.

La législation de la signature électronique est encore loin d'être au point malgré certains essais aux Etats-unis, plus particulièrement dans l'état d'Utah.

4.3 Conclusion

Nous avons parler dans ce chapitre de la légalité du chiffrement face aux lois sur les écoutes téléphoniques (déjà un peu développée au chapitre 1) et de la signature électronique. D'autres aspects légaux sont à peine effleurés comme les responsabilités des tiers de confiance, car ils sont en plein développement pour le moment. Nous ferons encore remarquer le besoin urgent de modifier la législation afin d'accepter les signatures électroniques certifiées par des tiers de confiance. Enfin, nous ferons remarqué que, n'étant pas juriste, le développement juridique est relativement restreint.

EXEMPLES DE TTP EXISTANT

5. *Exemple de TTP existant*

5.1 Interbank Standards Association Belgium

a) Introduction

Isabel est un tiers de confiance mis au point par les banques belges. Il ne certifie que des utilisateurs et pas d'autorités de certification subordonnées. Isabel est avant tout un logiciel multibancaire standardisé qui offre par le biais d'une procédure d'accès unique des fonctionnalités communes à toutes les banques belges. L'abonné d'Isabel peut effectuer toutes ses opérations d'electronic banking courantes avec les 30 banques d'Isabel. Il peut encore échanger du courrier électronique sécurisé, consulter des fournisseurs d'informations financières, commerciales et économiques et surfer sur Internet. En dehors de ces fonctions communes, l'abonné d'Isabel peut également recourir aux services privatifs proposés par chacune de ses banques. Isabel est encore une plate-forme qui intègre les standards de sécurité les plus sévères en matière de transmission de paiements et de certification des utilisateurs. A ce titre, Isabel applique le système de sécurité de la Smart Card. Cette technologie, basée sur le système RSA, garantit au moyen d'une seule carte et d'un seul mot de passe pour toutes les banques, la parfaite sécurité et confidentialité de toutes les opérations effectuées via le réseau d'Isabel.

b) Système d'enregistrement

La clef secrète d'un client n'est créée ni par Isabel, ni par une banque, ni par un autre fournisseur de services, mais par le client lui-même. Lorsqu'un client s'abonne à Isabel, une carte à puce vierge lui est remise. Le client peut alors initialiser la carte en utilisant un programme. Ce programme est un

algorithme qui combine une série de facteurs quasi-aléatoires pour créer la clef secrète et transférer cette clef secrète sur la smart Card. Toute trace de la clef secrète sur l'ordinateur est automatiquement effacée. Ce même programme calcule également la clef publique correspondante. Le seul endroit où se trouve la clef secrète est donc la carte à puce. Personne ne peut lire directement la clef secrète sur la carte et pour utiliser la carte à puce il faut lui attribuer un PIN (Personnal Identification Number).

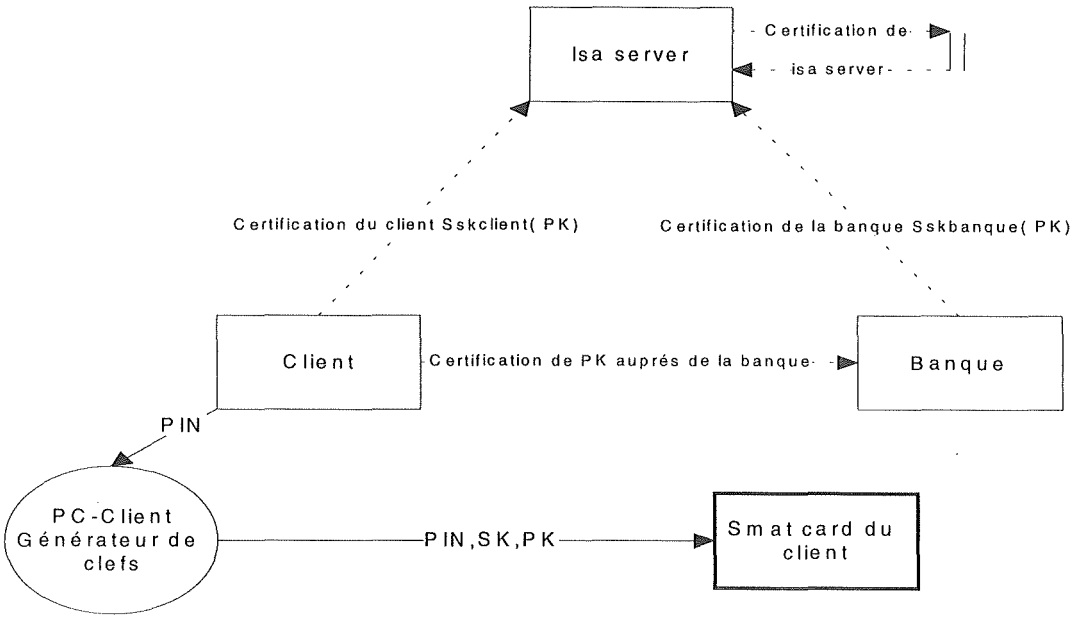

Figure 17. Système de certification Isabel

La certification de la clef publique du client s'effectue selon une procédure relativement complexe. Cette procédure consiste en trois demandes d'enregistrement (voir figure ci dessus), effectuées par trois parties indépendantes: le client, sa banque et Isaserver. Les trois demandes sont alors comparées; si tout est correct, un certificat est alors attribué. Ce certificat est en fait la signature électronique d'Isabel, apposée sur la clef publique du client.

c) L'utilisation des certificats

Les certificats sont ici utilisés de manière cachée pour des applications de gestion bancaire. Mais il existe aussi des fonctions de messagerie électronique

qui permettent de chiffrer et de signer des messages. Nous ne disposons pas de toutes les informations possibles mais nous pensons que les protocoles utilisés sont très semblables au protocole X.509 utilisé également par dans EDI (Echange de Données Informatiques). Isabel s'inspire fortement des normes d'EDI. Le système de messagerie suit la norme X400 (norme de communication reconnue internationalement). IsaNet est le réseau propre à Isabel qui fonctionne selon le protocole TCP/IP avec une capacité de 128 Ko/s extensible à 256 Ko/s. Il semblerait que la puissance actuelle le réseau soit relativement faible et ne pourrait pas gérer le courrier banquaire de grosses entreprisses (telle Canal+ Belgique où les message sont de l'ordre de 30 mégas bytes).

Tout message électronique doit contenir une adresse x400. Ces adresses sont reprises pour tous les abonnés d'Isabel dans un annuaire appelé répertoire X.500. La première fonction du X.500 est de proposer à tout moment la liste d'abonnés la plus récente. Pour chaque abonné on trouve:

- L'adresse X.400
- Les données concernant l'utilisateur Isabel (abonnement, identifiant d'utilisateur de réseau et identifiant de la boite aux lettres)
- Les coordonnés physiques (adresse, numéro de téléphone, fax.....)
- Le Certificat Isabel reprenant la clef publique qui permet de vérifier la signature électronique de l'abonné;

Nous soulignerons que nous n'avons pas trouvé de description des méthodes de répudiation des certificats Isabel.

5.2 BelSign

a) Introduction

Les services de certification publique de BelSign sont conçus pour supporter le commerce électronique sécurisé et d'autres services de sécurité générale dans le but de satisfaire les besoins techniques, commerciaux et personnels des utilisateurs en signatures digitales. Afin de réaliser ce but,

BelSign prend les rôles de tiers de confiance en créant, gérant, suspendant et révoquant des certificats. BelSign est prévu pour répondre à des demandes très variées du point de vue de la sécurité, notamment pour les algorithmes de chiffrements utilisés.

La génération du certificat par BelSign permet la confirmation de la relation entre une clef publique et l'identité, soit d'une autorité de certification, soit d'un utilisateur. Un haut niveau de gestion de ces certificats inclut la gestion de leurs enregistrements, leurs identifications, leurs authentifications, leurs générations, leurs révocations, leurs suspensions et la génération d'une trace. Les services de BelSign supportent une variété large de mécanismes de sécurités pour protéger les communications. BelSign fournit à ces utilisateurs un schéma qui permet aux autres parties d'utiliser ses services de sécurité (par exemple, quelqu'un qui n'est pas enregistré chez BelSign peut recevoir des messages signés).

BelSign offre trois niveaux distincts de certification. Selon le type de certificat désiré les utilisateurs peuvent postuler par E-mail, en écrivant à BelSign ou en se rendant dans une autorité d'enregistrement locale.

b) Les certificats « BelSign »

Il existe trois classes de certificats :

- certificats de classe 1 :Ils sont fournis à des individus. Leur seule sécurité est la confirmation que le nom et l'adresse E-mail du souscripteur sont bien identifiants dans le répertoire dépositaire des certificats. Ces certificats sont communiqués électroniquement au souscripteur. Ces certificats sont essentiellement utilisés pour faire du web browsing et du E-mailling en améliorant peu la sécurité de ces environnements. Ces certificats n'améliorent en aucune façon l'authentification du souscripteur.
- certificats de classe 2 : Ils sont fournis à des individus. Leur seule sécurité est la confirmation que les informations fournies par le souscripteur ne sont pas en conflit avec des informations contenues dans des bases de données existantes. Ces certificats sont utilisés pour des transactions peu risquées. BelSign propose un programme qui a pour objet de supporter la

validation software des certificats de classe 2. La preuve apportée par de tels certificats est raisonnable mais pas indiscutable.

- certificats de classe 3 : Ces certificats sont décernés à des entreprises et à des individus. Pour obtenir un de ces certificats, un individu doit se présenter devant une autorité d'enregistrement locale. Une organisation qui veut obtenir un certificat qui prouvera l'existence et le nom de l'organisation (qu'elle soit privée ou publique) choisira un certificat de classe 3. La validation de ces certificats pour les entreprises comprend les vérifications semblables à celle du niveau 2 et des vérifications par courrier physique (postal). Ce type de certificat est souvent pour des applications de commerce électronique comme l'électronique banking ou des applications orientées vers des services pour des membres uniquement.

Les certificats délivrés par BelSign se réclament de la norme X.509. En particulier, la norme X.509 v3 qui permet d'ajouter des extensions au certificat. Cette option a été fortement utilisée par BelSign qui a introduit plusieurs extensions pour améliorer les possibilités des certificats. La fonction de chaque extension est indiquée par une valeur identifiante standard. De plus, à chaque extension du certificat est associée une valeur critique (booléenne). Cette valeur est initialisée par BelSign sur base des informations reçues du souscripteur. Si cette valeur est à true pour l'extension d'un certificat, chaque utilisateur doit considérer ce certificat comme invalide si les informations attendues ne sont pas présentent dans l'extension. Si cette valeur est à false, l'utilisateur ne doit pas tenir compte de l'extension. Plusieurs extensions sont bien définies par la norme ISO/IEC9594-8 comme :

•l'extension des contraintes de base (qui définit le rôle et la position d'un certificat dans une chaîne de certification, voir figure ci-dessous.)

•l'extension d'utilisation de clefs (qui permet de restreindre l'utilisation des clefs)

•l'extension sur la politique de certification (qui pointe vers un URL où se trouve un document décrivant la politique de certification)

•l'extension d'identification approfondie.

BelSign a prévu de permettre à ses utilisateurs de définir des extensions supplémentaires privées pour adapter les certificats à leur environnement applicatif.

Les services offerts par BelSign ont été mis au point pour fonctionner dans une hiérarchie composée d'autres autorités de certification (certaines subordonnées à BelSign).

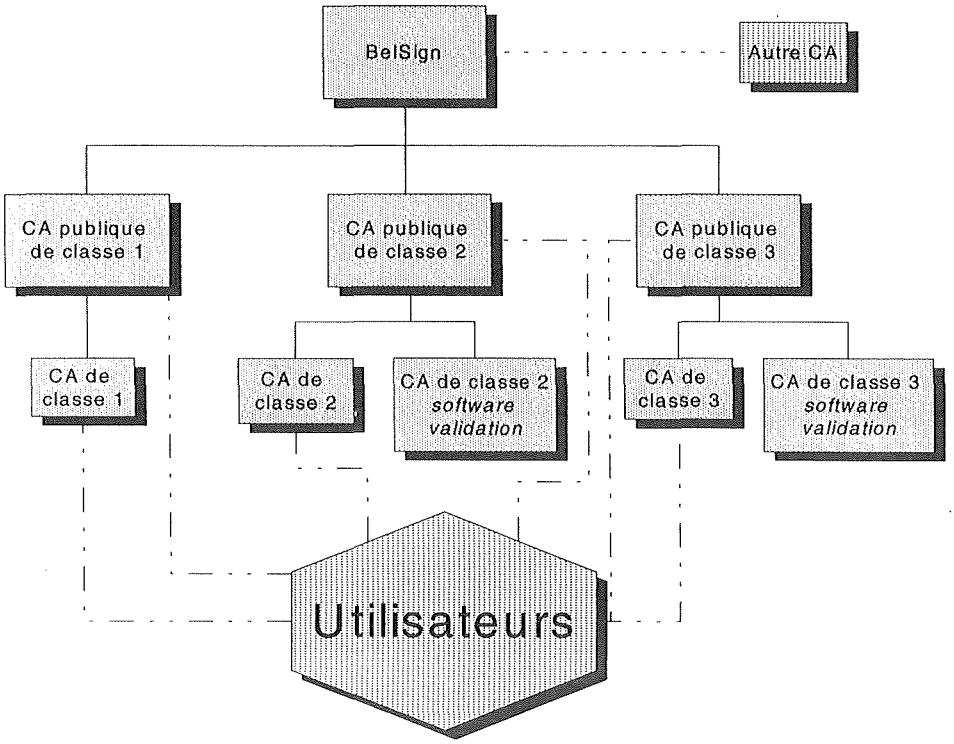

Figure 18 Hiérarchie d'une infrastructure à clef publique
L'autorité racine est représentée par BelSign. En dessous de ce niveau on peut remarquer des arbres d'autorité de certification. Mais à chaque niveau il y a la présence des utilisateurs qui peuvent interagir.

En plus de la structure hiérarchique représentée sur la figure ci-dessus, certaines autorités de certifications peuvent déléguer certaines fonctions d'enregistrement à des autorités locales d'enregistrements (ou chambres de commerce qui servent pour l'identification). Les autorités de certification fonctionnent en respectant la politique générale de sécurité de BelSign (BelSign CPS) pour la création, la gestion et la révocation des certificats des 3 classes. Le Repositary est un répertoire public qui permet l'utilisation du protocole X.509 en contenant une liste de certificats, une liste des certificats révoqués et d'autres informations.

c) Prérequis pour les opérations de certification

L'ensemble des responsabilités de BelSign est énoncé de façon contractuelle dans le « BelSign Certification Practice Statement (CPS)». BelSign s'engage à maintenir plusieurs services tels que l'enregistrement d'actions réalisées, la gestion du marquage temporel de différents documents (C.F. [BELS96] p20), la préservation des certificats pendant une période de 5 à 20 ans, le recouvrement des fonctionnalités en cas de désastre. BelSign doit également assurer la confidentialité de certaines informations et donc faire accréditer son personnel et le personnel d'autorité subalterne pour certaines tâches. De plus, chaque élément hardware ou software doit être accrédité par BelSign. BelSign utilise des cryptomodules pour toutes les opérations réclamant l'utilisation de ses clefs privées sauf pour des certificats de classe 1.

BelSign utilise le partage de secrets, pour améliorer la sécurité de ses clefs privées et permettre leur recouvrement en cas de désastre. Un second but du partage de secrets est de garantir « légalement » aux différentes entités que l'autorité de certification possède bien les clefs privées. Les secrets partagés sont remis à des gardiens de secrets. Une majorité de ces gardiens doivent être présents lors de la création et la distribution des secrets sur des éléments hardware. Ceux-ci acceptent leur rôle de gardien par un contrat qui spécifie leurs responsabilités.

Les autorités locales d'enregistrement occupent des positions de confiance dans le système. La formation des autorités de certification varie selon les types de certificat délivré. Elle est presque nulle pour les certificats de types un et deux et, elle est relativement approfondie pour des certificats de type trois.

En cas de mise hors service d'une autorité délivrant des certificats, le CPS prévoit certaines actions afin de limiter les problèmes. L'autorité doit prévenir 90 jours avant la révocation des certificats et trouver un arrangement pour préserver ses enregistrements. Elle peut aussi faire passer ses services à une autorité héritière.

d) Procédure d'enregistrement

Chaque utilisateur possède préalablement ses clefs privées ou les génère en utilisant un système sécurisé évitant les usages frauduleux (ne pas utiliser nécessairement le matériel fourni par BelSign). La protection de la clef publique dépend exclusivement du souscripteur.

Les informations fournies par un souscripteur sont différentes selon le type de certificats désirés. Nous ne reprendrons dans le tableau ci dessous que les informations obligatoires. Pour les autres informations le lecteur se référera à [BELS96]

Classe 1	Nom Clef publique Accord du souscripteur exécuter (preuve de la clef publique)
Classe 2	Adresse E-mail Nom légal Nom proposé (identifiant) Adresse Clef publique date de naissance Numéro de téléphone Phrase mot de passe pour authentifier plus tard le souscripteur information pour le paiement Accord du souscripteur exécuté [si tiers certifiant] garantie du software utilisé
Classe 3	idem que pour la classe 2 plus: Accord certifié par un notaire ou autorité locale d'enregistrement avec présence physique et présentation de trois preuves d'identité.

Tableau 2. Informations requises lors de la demande de certification

Après réception du formulaire, l'autorité de certification procède à plusieurs validations comme l'identité du souscripteur, le lien entre la clef privée et la clef publique, l'exactitude des informations contenues dans le

certificat. Page 34 du CPS on peut trouver un tableau reprennant différents niveaux de validation.

Les validations peuvent être faites par la présence physique des personnes dans le cas d'individus ou par la confirmation d'un tiers. Mais, dans ce dernier cas, la vérification se fait on line mais cela peut poser des problèmes législatifs dans certains pays car il y existe des lois sur la protection des données informatiques relatives à la vie privée. Dans le cas d'organisations, des tiers confirment le nom, l'adresse et d'autres informations. Les organisations peuvent également demander l'autorisation de devenir des tiers certificateurs. Les tiers fournissent également des numéros de téléphones afin de permettre des validations off-line. A partir des certificats de classe 2 la vérification de l'adresse postale se fait par l'envoi de référence par la poste.

e) La création et l'acceptation des certificats

La création d'un certificat indique qu'il y a eu une validation complète de la demande de certificat. Un certificat ne devient valide qu'après son acceptation par le souscripteur. BelSign ne peut pas produire un certificat sans le consentement du souscripteur. Par contre, BelSign se réserve le droit de refuser un certificat sans avoir à justifier sa décision. BelSign s'engage à ce qu'il n'y ait aucune erreur de transcription dans les certificats et que le certificat réponde aux attentes du CPS. La validité des certificats est limitée à deux semaines pour les certificats de classe 1 et à un an pour les autres.

L'acceptation du certificat par un souscripteur se fait on line via le web ou via E-mail. Par l'acceptation du certificat le souscripteur signifie son accord avec les informations contenues dans le certificat. Il s'engage à ce que chacune des signatures réalisées grâce à sa clef privée soit valide. Il est d'accord de ne jamais révéler sa clef privée et d'utiliser ses clefs dans la limite de la légalité. A l'acceptation du certificat, une copie de celui-ci est publiée dans le répertoire dépositaire de BelSign, mais le souscripteur peut le publier dans d'autres répertoires.

f) L'utilisation des certificats

La vérification d'une signature digitale est entreprise pour déterminer si la signature digitale a été créée par la clef privée correspondant à la clef publique listée dans le certificat du signataire et que le message n'a pas été endommagé depuis la création de la signature. Pour établir cette vérification, on vérifie si le déchiffrement du message ou de la hasch value est correcte. nous devons ensuite établir une chaîne de certificats pour la signature digitale qu'il faudra valider. Ensuite il faut vérifier si aucun de ces certificats ne se trouvent dans le répertoire de révocation.

Lors de la signature, le signataire doit indiquer l'heure et la date de signature. Une signature digitale doit être créée pendant la période de validité du certificat. Un document signé peut selon BelSign être considéré comme un papier signé. On peut utiliser les clefs pour la signature, mais également pour la confidentialité des messages en chiffrant avec la clef publique du destinataire.

g) Révocation, suspension et expiration d'un certificat

Les raisons de la révocation d'un certificat sont dans la plupart des cas le fait que la clef privée soit compromise, que le sujet du certificat n'a pas respecté ses engagements envers BelSign ou qu'un événement imprévisible a compromis la sécurité des informations d'un client de BelSign.

BelSign mettra fin à une suspension à la demande du sujet du certificat si celui-ci est bien identifié par BelSign et que la suspension était non fondée.

BelSign peut également révoquer un certificat à la demande du souscripteur ou s'il découvre que le certificat n'a pas été produit selon les règlements présents dans le CPS. Une fois un certificat révoqué, BelSign fera la publicité nécessaire à la révocation du certificat. Il devra notamment publier les certificats révoqués dans un répertoire particulier (le répertoire de révocation) et répondre aux demandes des utilisateurs touchés par la révocation en leur fournissant des certificats.

Pour les certificats arrivés à expiration BelSign ou l'autorité subalterne devra prévenir les sujets de ces certificats. Le processus de renouvellement

est le même que celui d'adhésion.

Les autorités de certifications subalternes auront les mêmes obligations

CONCLUSION

Conclusion

Il est maintenant temps de se pencher sur ce travail pour voir si nous avons atteint nos objectifs. Notre objectif principal était de permettre au lecteur de réaliser une infrastructure de tiers de confiance. Pour réaliser cet objectif, nous avons introduit les techniques de cryptographie communes à la plupart des tiers de confiance, comme le chiffrement asymétrique ou les cartes à puce. Nous avons également mis en évidence les différents rôles des tiers de confiance que nous avons développés . Ensuite, nous avons proposé plusieurs protocoles de communications sur lesquels peuvent se baser une infrastructure de tiers de confiance. Nous avons sélectionné un de ces protocoles. A partir de ce protocole, nous avons développé une infrastructure à tiers de confiance convenant pour une petite organisation telle que l'Institut. Le développement de cette infrastructure a été réalisé en langage courant et via le langage de spécification fonctionnelle vu au cours de Méthodologie de Développement de Logiciel de deuxième Licence.

Une fois tous ces outils mis dans les mains du lecteur, nous avons trouvé intéressant de développer des questions juridiques directement liées aux tiers de confiance. Nous avons donc parlé du problème du chiffrement face aux lois sur les écoutes téléphoniques et du problème de la validité des signatures électroniques. D'autres aspects légaux sont à peine effleurés comme les responsabilités des tiers de confiance, car ils sont en plein développement pour le moment. Nous ferons encore remarquer le besoin urgent de modifier la législation afin d'accepter les signatures électroniques certifiées par des tiers de confiance.

Nous avons enfin apporté des dernières informations pour le lecteur en reprenant deux exemples de tiers de confiance existant en Belgique. Ces tiers de confiance étant réalisés pour un environnement beaucoup plus ouvert, ils permettent au lecteur d'admettre une autre dimension des tiers de confiance.

Il est clair que notre travail est limité, pour plusieurs raisons. Une première raison est qu'il ne pourrait être exhaustif, car la quantité de production dans ce domaine est immense. Il aurait été, de plus, intéressant

de mener à bien l'implémentation de l'infrastructure développée dans ce travail. Une autre faiblesse de ce travail vient du fait que nous n'avons considéré que le niveau logique des communication. Cette implémentation a en partie été réalisée à partir de Delphi et de PGP, mais n'a pas pu être terminée. Enfin, nous ferons remarqué que, l'auteur n'étant pas juriste, le développement juridique est relativement restreint. Mais fi de ces observations, nous pensons, par ce travail, avoir apporté la plupart des informations nécessaires à la réalisation d'une infrastructure de tiers de confiance.

Soulignons encore les quelques apports plus personnels à ce travail. Un premier apport a été effectué dans le choix des rôles des tiers de confiance que nous trouvons en partie dans la littérature mais pas sous la forme de taxinomie présentée dans ce travail. Un autre apport personnel réside encore dans la mise au point du protocole à clef de session « amélioré » et de l'infrastructure s'y rattachant.

Les tiers de confiance ont sans aucun doute un avenir très prometteur. Il suffit de voir le nombre de projets qui vont en ce sens que ce soit dans le domaine commercial, médical ou administratif (ex. Projet AGORA). Beaucoup d'organisations visent la confidentialité et l'authentification de leurs communications sur des réseaux ouverts. Comment répondre à ce besoin mieux que par la mise au point d'une infrastructure de tiers de confiance.

BIBLIOGRAPHIE

Bibliographie

Cryptographie

[BRIG88]. Roy Bright; « Smart cards principles, practice,applications »; ELLIS Horwood books in information technology ; 1988

[AZIZ96],Ashar Aziz, Tom Markson, Hemma Prafullchandra; « Skip-x509 »; IPSEC Working Group INTERNET_DRAFT;Sun Microsystems (Aout 1996)

[ELGA85] Taher Elgamal; « A public key cryptosystem and a Signature Scheme based on discrete logarithms » ; IEEE Transactions on information theory ,juillet 85,VOL 31 NUM 4 469-472

[DENN82] Dorothy E.Denning; « Cryptographie and data sécurity »; Addison-Wesley Publishing company isbn 0-201-10150-5)

[DENN96] Dorothy E.Denning and Dennis K.Brandstad; « A taxonomie for a key escrow systems »; Communication of the ACM, Mars 1996, VOL 39 NUM 3 34-39

[DIFF] Whitfield Diffie & Martin E.Hellman; « New directions in Cryptographie »; IEEE Transactions on information Theory, Novembre 1976, VOL22 NUM 6

[LEIN 90] Lein Harn & Hung-yu Lin ; « A Cryptographic key generation scheme for multilevel data security »; Computer & security, 9 (1990) 539-546

[GARF96] Garfinkel Simso L. « Public key cryptography » ;Computer ;Rom Vetter,North Dakota state university, juin 96 pp101-105

[GUEUL93] Patrick Gueulle ; « Cartes à puce Initiation et applications » ;ETSF (edition techniques et scientifiques) Paris; aout 1993

[HELL79];M.E. Hellman; « The mathématics of publickey cryptography »; scientific american; v.241, n.8, Aug 1979; pp146-157

[PEER96] ; Michel Peermans ; « FAST First Attempt to secure Trade » ;National fundation of commerce and industry ; juin 1996

[MAHE96] Devid Paul Maher; « CryptoBackup and key escrow »; Communication of the ACM, Mars 1996, VOL 39 NUM 3 48-53

[RIVE 78] Rivest, Shamir & Adleman, « A Method for Obtaining Digital signatures and Public-key cryptosystems »; Communication of ACM fevrier 1978 VOL 21 NUM 2

[RIVE 88] Ronald Rivest; «Cryptology » ;.Draft of January 21-1988 handbook of theorical computer Sciences, Chap 13 ,pp30

[SCHN93] Bruce Schneier ; « Applied Cryptography »;John Wiley & sons, inc.; New York1993

[SHAM82] A. Shamir; « A polynomial Time algorithm for breaking the basic Merkle-Hellman cryptosystem »; Advances in cryptology: Proceedings of crypto82, plenum press, 1983, pp 339-340

[WALK96] Stephen B.Lipner, Carl M Ellison, and David M Balenson; « Commercial Key Recovery »; Communication of the ACM, Mars 1996, VOL 39 NUM 3 41-48

[WALL90] Dominique de waleffe & Jean-jacques Quisquater« Corsair: A Smart Card for public Key cryptosystems »; Proc. of Crypto'90

[ZORE94]; José Luis Zoréda & José Manuel Otón; « Smart cards » ; Edition Artech House Norwood,1994

Notion de droit

[DAVI 97] Etienne Davio ; « Questions de certification signature et cryptographie »; Cahiers du Centre de Recherches informatique et droit, 1997 , VOL 1 NUM 12 67-86

[FROO 96] Michael Froomkin ; « The essential role of trusted third parties in electronic commerce »; 75 Oregon L.Rev.49 (octobre 96) Site web:http://viper.law.miami.edu.~froomkin:/articles:trustdno.htm

[VERT94] .....;« Livret vert sur la sécurité des systémes d'information »; ;Version 4.2.1;février 1994

[MIRI97] Antoine Mireille; « Les aspects Juridiques de la sécurité de l'information (Ecoutes, prise de connaissance et enregistrement de communications et télécommunications privées et usage de la cryptographie en droit belge) »; BELINFOSEC - groupe juridique - rapport final; janvier 1997

[PARI96] Serge Parisien & Pierre Trudel; « L'identification et la certification dans le commerce électronique (Droit, sécurité, audit et technologies) »; Edition Yvons Blais ;mars 1996

Protocole de communication

[RAMA 90] Raju Ramaswamy; « A key management algorythm for secure communication in open systems interconnection architecture »; Computer & security, 9 (1990) 77-84

[LAUN 92] Rotraut Laun; « Asymmetric User Authentification »; Computer & security, 11 (1992) 173-183

[CART 95] Glyn Carter; Smith; « Paper on TESTFIT for TEDIS TTP workshop »; System Engineering Ltd (1995)

[NEEDH 78] Roger M.Needham & Michael D. Shroeder; « Using Encryption for Authentication in large network of computers »; Communication of the ACM, VOL21 NUM 12 (1978) 993-999

Question générale sur les TTP

[BELS96] BelSign certification practice statement; adresse electronique: www.BelSign.be/repository/CPS

[ISAB96]